

Dr. Sancho Salcedo Sanz y Dr. José Antonio Portilla Figueras, Profesores Titulares de Universidad del Área de Conocimiento de Teoría de la Señal y Comunicaciones de la Universidad de Alcalá,

#### CERTIFICAN

Que la tesis **“Técnicas de soft-computing para el desarrollo de redes de acceso móvil con control de polución electromagnética”**, presentada por Dña. María Pilar García Díaz, realizada en el Departamento de Teoría de la Señal y Comunicaciones bajo nuestra dirección, reúne méritos suficientes para optar al grado de Doctor, por lo que puede procederse a su depósito y defensa.

Alcalá de Henares, septiembre de 2013.

Fdo. Dr. D. Sancho Salcedo Sanz

Fdo. Dr. D. José Antonio Portilla Figueras



Dña. María Pilar García Díaz ha realizado en el Departamento de Teoría de la Señal y Comunicaciones y bajo la dirección de los Doctores D. Sancho Salcedo Sanz y D. José Antonio Portilla Figueras, la tesis doctoral titulada **“Técnicas de soft-computing para el desarrollo de redes de acceso móvil con control de polución electromagnética”**, cumpliéndose todos los requisitos para la tramitación que conduce a su posterior lectura.

Alcalá de Henares, septiembre de 2013.

EL DIRECTOR DEL DEPARTAMENTO

Fdo. Dr. D. Saturnino Maldonado Bascón







ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE TEORÍA DE LA SEÑAL Y COMUNICACIONES

Tesis Doctoral

TÉCNICAS DE SOFT-COMPUTING PARA EL  
DESARROLLO DE REDES DE ACCESO MÓVIL CON  
CONTROL DE POLUCIÓN ELECTROMAGNÉTICA

Autor:

María Pilar García Díaz

Directores:

Dr. Sancho Salcedo Sanz

Dr. José Antonio Portilla Figueras

**Septiembre, 2013**



# Abstract

This Ph. D Thesis is focused on The Mobile Network Deployment Problem (MNDP) a class of assignment problem in GSM networks. It is usually stated as an optimization problem which looks for a solution with the minimal possible economic cost of the deployment and the maximal possible coverage percentage of the communication network. Thus, in the classical problem, the objective function is usually defined by both variables: cost and coverage percentage. In this work we include a third variable in the objective function: the electromagnetic radiation due to the base stations working. The society feels some apprehension because the possible negative effects of radiation in human life. The idea of considering the pollution control in the MNDP came up from this social feeling.

This new optimization problem is solved through two methods: an evolutive algorithm and a new algorithm recently published: The Coral Reefs Optimization (CRO). It is bio-inspired on the coral reefs behaviour in natural world. The results of both algorithms have been compared with other three algorithms considering the same fitness function. They are: the Particle Swarm Optimization algorithm, the Harmony Search algorithm and a greedy one. Our experiments show that CRO is the best algorithm to solve the MNDP.



# Resumen

Este trabajo de Tesis Doctoral estudia el problema del despliegue de redes móviles (*Mobile Network Deployment Problem* o MNDP), orientado a la localización de estaciones base en una red de telecomunicación GSM. Tradicionalmente, este problema de optimización consiste en hallar una solución tal que, con el mínimo coste económico de la red, asegure un grado de servicio mínimo en la zona. Así la función de evaluación maneja dos variables: el coste y el grado de cobertura de la red en el área de estudio. Una de las aportaciones de este trabajo es la incorporación de una nueva variable a dicha función: la radiación electromagnética sobre el terreno en el que opera la red. Existen numerosos estudios que abordan el problema a partir del coste y el grado de servicio, sin embargo no hemos encontrado investigaciones que persigan minimizar la cantidad de radiación emitida por las estaciones base. La sociedad actual mantiene cierta aversión a la radiación que emiten los equipos de telefonía móvil. De este sentimiento surge la idea de incorporar el parámetro de polución electromagnética al problema de optimización MNDP.

El problema se aborda mediante métodos metaheurísticos de optimización: un algoritmo evolutivo tradicional, y un novedoso algoritmo recientemente publicado, el *Coral Reefs Optimization* (CRO). Este último es un algoritmo bio-inspirado que se basa en la simulación de los procesos que de los arrecifes de coral. Los resultados obtenidos de la aplicación de ambas metodologías al problema MNDP han sido comparados con otros tres algoritmos metaheurísticos con la misma función de evaluación. Estos son: el algoritmo *Particle Swarm Optimization*, el *Harmony Search* y un algoritmo tipo *greedy*. Los experimentos realizados sitúan, de manera ampliamente diferenciada, el algoritmo CRO como el más apropiado para resolver el problema MNDP.



Dedicado a César Fernández Arroyo





## Agradecimientos

En primer lugar, quiero agradecer a los directores de esta Tesis Doctoral: Dr. Sancho Salcedo Sanz y Dr. José Antonio Portilla Figueras, por haber recogido mi participación en este cometido. También quiero darles las gracias por el tiempo dedicado y su labor de seguimiento. Obviamente, sin ellos no hubiera sido posible ni comenzar, ni realizar ni finalizar este trabajo de investigación.

Añado mi agradecimiento a los investigadores y compañeros del proyecto de investigación “Estudio de las emisiones radioeléctricas en las bandas de frecuencias asignadas a los servicios de telefonía móvil, en la ciudad de Alcalá de Henares” por la cesión de datos de dicho proyecto. Éstos han sido de gran utilidad para la elaboración de la Tesis Doctoral. Especialmente quiero agradecer al Dr. Pablo López Espi y a la Dra. Rocío Sánchez Montero su colaboración y siempre abierta disposición.

Deseo reflejar aquí mi agradecimiento al Dr. Francisco López Ferreras, al Dr. Manuel Rosa Zurera y al Dr. Saturnino Maldonado Bascón, por su apoyo como directores del Departamento de Teoría de la Señal y Comunicaciones desde mi incorporación al mismo. A pesar de sus cargadas agendas, siempre me recibí de ellos el tiempo necesario y determinadas recomendaciones por las que reitero mi gratitud.

Extiendo mi agradecimiento a los miembros del departamento, en especial al grupo docente en el que trabajo, por la ayuda personal que me ofrecieron siempre que fue posible. Entre ellos tengo la suerte de contar con los compañeros la Dra. Silvia Jiménez Fernández, el Dr. Pedro Amo López y el Dr. Rafael Cambralla Diana. Deseo mencionar aquí también a miembros, actuales o pasados, del grupo de investigación al que pertenezco, como el Dr. Leopoldo Carro Calvo, el Dr. Emilio Gedeón Ortiz-García y al ingeniero Javier Plaza Laina, a los que doy las gracias por su colaboración y ayuda en ciertos momentos.

Finalmente, y no por ello menos importante, tengo mucho que agradecer a mi marido, a mi madre y a mi padre por su colaboración en el ámbito familiar. Sin su ayuda, hubiera resultado francamente imposible realizar el trabajo. **Muchas gracias.**



# Contenido

<b>Contenido</b>	<b>V</b>
<b>Índice de figuras</b>	<b>IX</b>
<b>Índice de tablas</b>	<b>XIII</b>
<b>Índice de algoritmos</b>	<b>XIV</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Introducción . . . . .	1
1.2 Radiación electromagnética y límites de exposición . . . . .	5
1.3 Estudio de las emisiones radioeléctricas en la ciudad de Alcalá de Henares . . . . .	10
1.4 Objetivos y estructura de la Tesis Doctoral . . . . .	13
<b>2 Estado del arte</b>	<b>17</b>
2.1 Técnicas de soft-computing . . . . .	17
2.2 Introducción a los algoritmos genéticos . . . . .	21
2.2.1 Codificación de la solución . . . . .	23
2.2.2 Función de evaluación . . . . .	26
2.2.3 Selección de individuos . . . . .	26
2.2.3.1 Método de selección por ruleta . . . . .	28
2.2.3.2 Sobrante estocástico . . . . .	29
2.2.4 Operador de cruce . . . . .	30
2.2.5 Operación de mutación . . . . .	33
2.2.6 Búsqueda local . . . . .	38
2.3 Introducción al algoritmo PSO . . . . .	39
2.3.1 Descripción de un algoritmo PSO . . . . .	39
2.4 Introducción al algoritmo Harmony Search . . . . .	43
2.5 Aplicaciones de técnicas de heurísticas en problemas de localización de estaciones base en redes de telecomunicación . . . . .	47
<b>3 Planteamiento matemático del problema MNDP</b>	<b>49</b>
<b>4 Métodos desarrollados para la resolución del problema MNDP</b>	<b>55</b>
4.1 Algoritmo evolutivo desarrollado para el problema MNDP . . . . .	55
4.1.1 Codificación empleada . . . . .	55
4.1.2 Proceso de comienzo del algoritmo . . . . .	56
4.1.3 Proceso de selección de la población . . . . .	58
4.1.4 Proceso de selección de progenitores . . . . .	59
4.1.5 Operador de cruce o recombinación . . . . .	59

4.1.5.1	Operador de cruce TBC . . . . .	61
4.1.5.2	Operador de cruce DMC . . . . .	62
4.1.6	Operador de mutación . . . . .	63
4.1.7	Proceso de búsqueda local . . . . .	64
4.1.8	Función de evaluación . . . . .	69
4.1.9	Condiciones del parada . . . . .	71
4.2	Algoritmo <i>Greedy</i> desarrollado para el problema MNDP . . . . .	71
4.3	Algoritmo PSO desarrollado para el problema MNDP . . . . .	72
4.3.1	Conversión del espacio de soluciones continuo al discreto . . . . .	73
4.3.2	Codificación empleada . . . . .	74
4.4	Algoritmo HS desarrollado para el problema MNDP . . . . .	75
<b>5</b>	<b>Algoritmo Coral Reefs Optimization</b>	<b>79</b>
5.1	Arrecifes de coral en la naturaleza . . . . .	79
5.2	Estrategia del algoritmo Coral Reefs Optimization . . . . .	81
5.2.1	Parámetros específicos del algoritmo CRO . . . . .	82
5.2.2	Proceso de comienzo del algoritmo . . . . .	83
5.2.3	Métodos de reproducción del coral . . . . .	84
5.3	Algoritmo CRO desarrollado para el problema MNDP . . . . .	85
5.3.1	Proceso de comienzo del algoritmo . . . . .	86
5.3.2	Proceso de eliminación de réplicas . . . . .	86
5.3.3	Métodos de reproducción de corales . . . . .	87
5.3.3.1	Método de reproducción sexual . . . . .	88
5.3.3.2	Método de reproducción asexual . . . . .	89
5.3.4	Método de establecimiento en el arrecife . . . . .	90
5.3.5	Proceso de depredación en el algoritmo . . . . .	90
<b>6</b>	<b>Resultados</b>	<b>93</b>
6.1	Especificaciones del problema MNDP en un caso real . . . . .	93
6.2	Especificaciones para el EA . . . . .	97
6.3	Especificaciones para el algoritmo <i>Greedy</i> . . . . .	99
6.4	Especificaciones para el algoritmo PSO . . . . .	99
6.5	Especificaciones para el algoritmo Harmony Search . . . . .	101
6.6	Especificaciones para el algoritmo CRO . . . . .	101
6.7	Resultados con los algoritmos desarrollados . . . . .	103
6.7.1	Resultados obtenidos con el EA . . . . .	103
6.7.2	Resultados obtenidos con el algoritmo <i>Greedy</i> . . . . .	106
6.7.3	Resultados obtenidos con el algoritmo PSO . . . . .	107
6.7.4	Resultados obtenidos con el algoritmo HS . . . . .	109
6.7.5	Resultados obtenidos con el algoritmo CRO . . . . .	110
6.8	Comparación de las soluciones encontradas . . . . .	111
<b>7</b>	<b>Conclusiones</b>	<b>117</b>
7.1	Aportaciones de la Tesis Doctoral . . . . .	117
7.2	Líneas de trabajo futuras . . . . .	118
7.3	Lista de publicaciones . . . . .	119
<b>A</b>	<b>Tipos de radiación electromagnética</b>	<b>121</b>

<b>B</b>	<b>Algoritmo genético de agrupamiento aplicado al problema MNDP con restricción de capacidad</b>	<b>123</b>
B.1	Planteamiento matemático del problema MNDP con restricción de capacidad . .	123
B.2	Algoritmo genético de agrupamiento desarrollado para el problema MNDP . . . .	124
B.2.1	Codificación del problema . . . . .	125
B.2.2	Generación de la población inicial . . . . .	125
B.2.3	Operador de cruce o recombinación . . . . .	126
B.2.4	Operador de mutación . . . . .	128
B.2.5	Proceso de reparación de un individuo . . . . .	131
B.2.6	Proceso de búsqueda local . . . . .	132
	<b>Bibliografía</b>	<b>135</b>



# Índice de figuras

1.1	Evolución del número de líneas móviles, en millones de líneas, en España durante los años 2000-2011. Fuente: Comisión del Mercado de las Telecomunicaciones ( <i>CMT</i> ). . . . .	1
1.2	Grado de penetración de los principales servicios de telecomunicación en España a fecha el mes de diciembre de 2011, en número de líneas por cada 100 habitantes. Fuente: <i>CMT</i> . . . . .	2
1.3	Tráfico medio estimado de banda ancha móvil por tipo de dispositivo en España para 2016 (MB/mes). Fuente: <i>CMT</i> . . . . .	2
1.4	Evolución del número de estaciones base implantadas según la tecnologías GSM, GPRW y UMTS en España durante la década 2002-2011. Fuente: <i>CMT</i> . . . . .	3
1.5	Agrupación de los ocho distritos que componen la ciudad de Alcalá de Henares. Cada uno de ellos queda circunscrito por una línea de color granate. . . . .	12
1.6	Mapa de la ciudad de Alcalá de Henares, sobre la que se ubican de forma manual 509 puntos de medida de emisión radioeléctrica. . . . .	13
1.7	Mapa de emisiones radioeléctricas sobre la ciudad de Alcalá de Henares en el año 2006. . . . .	14
2.1	Clasificación de métodos heurísticos. Este trabajo se centra especialmente en los algoritmos genéticos, PSO y HS. . . . .	19
2.2	Diagrama de flujo básico de un algoritmo genético. En cada generación se realizan operaciones de evaluación o cómputo de la función de <i>fitness</i> , selección, cruce y mutación. . . . .	23
2.3	Ejemplo de codificación binaria. La codificación de la solución se conoce como genotipo. El individuo o cromosoma se compone de un número determinado de genes. El fenotipo se corresponde con la solución que representa el cromosoma. . . . .	24
2.4	Distribución de probabilidades de selección por el método de la ruleta según el ejemplo de la Tabla 2.1. . . . .	29
2.5	Ejemplo de la operación de cruce o recombinación con corte de punto único en un algoritmo genético. Cada cromosoma se fragmenta en dos partes y el individuo descendiente recibe un fragmento de cada progenitor. . . . .	30
2.6	Ejemplo de la operación de cruce uniforme en un algoritmo genético. El cromosoma descendiente se crea tomando los genes de uno u otro progenitor según indique la máscara binaria. . . . .	31
2.7	Ejemplo, en un algoritmo genético, de la operación de cruce por punto único con permutación aleatoria asociada. Los cromosomas progenitores sufren la misma permutación aleatoria antes del cruce. Tras realizar el cruce, se aplica la permutación inversa al descendiente. (*) Indica que el cromosoma está permutado. . . . .	32

2.8	Ejemplo de cruce de tipo <i>partial map crossover</i> en un algoritmo genético. El cromosoma descendiente hereda fragmentos completos de uno de los dos progenitores, completando sus huecos con los genes todavía ausentes y en el orden de aparición en el segundo de los progenitores. . . . .	33
2.9	Ejemplo de mutación binaria sobre un cromosoma. El contenido de los genes del cromosoma ubicados en las posiciones donde la máscara contiene “1” invierten su contenido, es decir los “1” se convierten en “0” y viceversa. . . . .	34
2.10	Ejemplo de mutación binaria uniforme sobre un cromosoma de codificación binaria y con umbral de probabilidad $threshold = 0,4$ . El vector de números reales aleatorios junto con el umbral de probabilidad generan la máscara binaria que se aplica sobre el cromosoma a mutar. . . . .	35
2.11	Ejemplo de mutación al borde en un algoritmo genético con codificación no binaria. El gen número 2 es seleccionado al azar para que su contenido (7,4) sea modificado, eligiendo uno de los dos valores extremos del intervalo disponible [1-10]. En este caso se ha tomado el valor límite superior. . . . .	35
2.12	Ejemplo de mutación de salto en codificaciones no binarias. Tras elegir al azar el gen a mutar cuyo valor es 7,4 y considerando una dirección ascendente hacia el máximo valor del intervalo [1-10], el gen es modificado por valor aleatorio entre su valor inicial y el máximo posible. . . . .	36
2.13	Ejemplo mutación de intercambio o <i>swap mutation</i> en un algoritmo genético con codificación basada en el orden. Dos genes con contenidos (7,4) y (2,7) intercambian sus posiciones. . . . .	37
2.14	Ejemplo de mutación de intercambio con el adyacente en codificaciones basadas en el orden para un algoritmo genético. El segundo gen (7,4) es seleccionado al azar y pasa a ocupar la posición de su adyacente (6,2). A su vez este último ocupa la posición del segundo. . . . .	37
2.15	Ejemplo de mutación de inversión en un algoritmo genético, un conjunto de genes consecutivos intercambian sus valores dos a dos comenzando por los genes más alejados entre sí. . . . .	38
2.16	Ejemplo de mutación con desplazamiento en una algoritmo genético para codificaciones basadas en el orden. Definido un conjunto de genes consecutivos, el contenido de estos se desplaza una posición a la derecha en el cromosoma. . . . .	38
2.17	Ejemplos de grupos de animales o <i>swarms</i> en la naturaleza (aves en la parte izquierda y peces en la imagen derecha de la figura). Cada animal decide su movimiento en base a su propio conocimiento y a su experiencia dentro del grupo. . . . .	40
2.18	Movimiento desde la iteración $k$ a la iteración $(k+1)$ de la partícula $i$ en el espacio de búsqueda de soluciones. A partir de su posición en la iteración $k$ se determina la posición en $(k+1)$ bajo el modelo de las Ecuaciones 2.8 y 2.10. . . . .	43
2.19	Analogía entre una orquesta de jazz compuesta por tres músicos probando un nuevo acorde (Do, Mi, Sol) y la evaluación de la solución ( $f(x_1, x_2, x_3) = f(1, 3, 5)$ ) en un problema de optimización. Figura tomada de [68]. . . . .	44
2.20	Ejemplo del uso de la <i>harmony memory</i> (HM). Analogía entre una orquesta de jazz con tres acordes aprendidos y el conjunto de tres soluciones encontradas para un problema de optimización de energía. Figura tomada de [68]. . . . .	45
4.1	Ejemplo de distribución de probabilidades de selección de individuos por el método de la ruleta para la población de la Tabla 4.1. . . . .	60
4.2	Ejemplo de cruce de tipo TBC de dos individuos en el EA. . . . .	62



4.3	Ejemplo de cruce de tipo DMC de dos individuos en el EA. . . . .	63
4.4	Ejemplo de mutación de un individuo para el EA. . . . .	64
4.5	Representación gráfica de un individuo del problema MNDP sobre el que se aplica el proceso de búsqueda local, identificando tres estaciones base prescindibles destacadas en color rojo. . . . .	66
4.6	Representación gráfica del individuo de la Figura 4.5 tras retirar de la solución las tres estaciones base identificadas como prescindibles. . . . .	67
4.7	Ejemplo de cómo la búsqueda local sobre un mismo individuo ofrece más de un resultado. Dependiendo del orden en el que se revisan las estaciones base de la solución al problema MNDP representada en la Figura 4.7(a), una de las dos estaciones base destacadas en color rojo resulta prescindible, pero no las dos a la vez. Las Figuras 4.7(b) y 4.7(c) muestran la solución final tras eliminar una u otra estación. . . . .	68
4.8	Representación de la matriz de cobertura $N \times M$ de un individuo como posible solución para una zona de estudio $A$ definida por 21 localizaciones como objetivo de demanda ( $\#P' = M = 21$ ). La solución consta de 7 estaciones base ( $N=7$ ). La matriz simboliza con puntos negros aquellos elementos $(i,j)$ donde la localización $j$ se encuentra dentro de la zona de cobertura de la estación base $i$ . . . . .	69
4.9	Ejemplo de codificación para el algoritmo PSO de una solución al problema MNDP a partir de la codificación empleada en el algoritmo evolutivo. . . . .	75
4.10	Ejemplo del contenido de la <i>harmony memory</i> (HM) para el problema MNDP. . . . .	77
4.11	Ejemplo de la operación de <i>memory consideration</i> a partir de la HM de la Figura 4.10. . . . .	77
5.1	Fotografías de diferentes corales en la naturaleza obtenidas a través de la revista National Geographic. . . . .	79
5.2	Esquema básico de la estructura de un pólipo de coral. Figura realizada por el Fulton-Montgomery Community College ( <a href="http://faculty.fmcc.suny.edu/">http://faculty.fmcc.suny.edu/</a> ). . . . .	80
5.3	Representación gráfica del estado de un arrecife bidimensional en el instante de inicio del algoritmo CRO. El color blanco señala los huecos libres que hay en la estructura para la siguiente generación. . . . .	83
5.4	Ejemplo de la evolución de un arrecife bidimensional en el algoritmo CRO durante $N$ generaciones. . . . .	84
5.5	Fotografías instantáneas de tipos de reproducción en corales duros. Fuente de las imágenes: (a) <a href="http://www.gbundersea.com">www.gbundersea.com</a> ; (b) <a href="http://vcebiology.edublogs.org">http://vcebiology.edublogs.org</a> . . . . .	85
6.1	Características de la mejor solución ofrecida por el algoritmo EA: (a) representación gráfica del individuo, (b) evolución de la función de evaluación de la simulación que genera la solución, (c) incremento de campo eléctrico para este individuo, (d) coste de la red y (e) porcentaje del área de estudio $A$ que queda sin servicio. . . . .	105
6.2	En el algoritmo PSO, características del mejor individuo obtenido mediante 60 simulaciones de poblaciones de tamaño 50 individuos: (a) representación gráfica de la solución, (b) evolución de la función de evaluación durante 200 iteraciones de la simulación que entrega el mejor individuo, (c) incremento de campo eléctrico para esta solución, (d) coste de la red y (e) porcentaje del área de estudio $A$ sin cobertura en este individuo. . . . .	108

6.3	En el algoritmo HS, evolución durante 9800 iteraciones de las tres variables de la función de <i>fitness</i> : 6.3(c) incremento de campo eléctrico, 6.3(d) coste de la red y 6.3(e) porcentaje del área de estudio <i>A</i> sin cobertura. . . . .	112
6.4	En el algoritmo CRO, evolución durante 200 iteraciones de las tres variables de la función de <i>fitness</i> : 6.4(c) incremento de campo eléctrico, 6.4(d) coste de la red y 6.4(e) porcentaje del área de estudio <i>A</i> sin cobertura. . . . .	113
6.5	En el algoritmo CRO, evolución de la ocupación del arrecife coral durante 200 iteraciones en cinco periodos equiespaciados de cuatro iteraciones consecutivas cada uno de ellos. . . . .	114
6.6	Representación gráfica de los mejores individuos ofrecidos por los algoritmos desarrollados (ver Tabla 6.8 para mayor detalle): algoritmo <i>Greedy</i> en (a); algoritmo PSO en (b); algoritmo de búsqueda armónica en (c) y (d); algoritmo evolutivo en (e) y (f); y algoritmo CRO en (g) y (h). . . . .	115
6.7	Evolución de la función de evaluación (valor medio en color azul y valor mínimo para el mejor individuo en color rojo) en los diferentes algoritmos: (a) algoritmo PSO; (b) algoritmo HS; (c) algoritmo EA; (d) algoritmo CRO. . . . .	116
A.1	Espectro de radiación electromagnética. Puede observarse que la radiación producida por las aplicaciones de radiocomunicación es de tipo no ionizante. Fuente: <a href="http://myscifi.org">myscifi.org</a> . . . . .	122
B.1	Ejemplo de codificación de una solución para el problema MNDP con el algoritmo GGA. . . . .	126
B.2	Ejemplo de cruce de dos individuos en el algoritmo GGA para el problema MNDP. . . . .	127
B.3	Ejemplo simplificado de la función del operador de cruce o recombinación en el algoritmo GGA. Representación gráfica de las soluciones de la Figura B.2, donde a partir de dos individuos aleatorios B.3(a) y B.3(b) se genera un nuevo individuo <i>offspring</i> B.3(c). . . . .	129
B.4	Ejemplo de la operación de mutación sobre una solución del problema MNDP con el algoritmo GGA. La solución previa a la mutación, formada por nueve estaciones base, se representa en la Figura B.4(a). El resultado de la mutación es la solución de la Figura B.4(b). Tras las operaciones de reparación y búsqueda local, se obtiene la solución de la Figura B.4(c), constituida por siete estaciones. . . . .	130
B.5	Ejemplo de mutación no eficaz sobre la parte de asignación del individuo. La Figura B.5(a) muestra una solución previa a la mutación. La Figura B.5(b) es el resultado inmediato de la mutación. Después de verificar las restricciones de cobertura y capacidad se obtiene la solución de la Figura B.5(c), que es notablemente parecida a la solución de la Figura B.5(a). . . . .	131

# Índice de tablas

1.1	Restricciones básicas de $SAR$ (W/kg) según la ICNIRP y la FCC en exposiciones a campos electromagnéticos en el rango de frecuencias [100 KHz - 6 GHz] con un tiempo medio de exposición de 6 minutos. Las medidas de $SAR$ en zonas parciales del cuerpo se refieren a 10 gr de tejido contiguo. . . . .	8
1.2	Niveles de referencia: Valores máximos de $E$ , $H$ y $S$ establecidos por los diferentes organismos para la frecuencia de 900 MHz en exposición controlada (ocupacional) y no controlada (público general). Nota: los organismos en rojo indican márgenes de obligado cumplimiento en Europa y EEUU. . . . .	9
1.3	Niveles de referencia: Valores máximos de $E$ , $H$ y $S$ para la frecuencia de 1800 MHz, establecidos por los diferentes organismos en exposición controlada y no controlada. Nota: los organismos en rojo indican márgenes de obligado cumplimiento en Europa y EEUU. . . . .	9
1.4	Niveles de referencia: Valores máximos de $E$ , $H$ y $S$ establecidos por los diferentes organismos para la frecuencia de 2100 MHz en exposición controlada y no controlada. Nota: los organismos en rojo indican márgenes de obligado cumplimiento en Europa y EEUU. . . . .	10
1.5	Valores máximos de $S$ establecidos por los diferentes organismos para la frecuencia de 2100 MHz en exposición no controlada. (*) agrupa la mayoría de los países miembros de la Unión Europea: Alemania, Austria, Dinamarca, Eslovaquia, Eslovenia, España, Estonia, Finlandia, Francia, Grecia, Holanda, Australia y Nueva Zelanda. . . . .	10
1.6	Valores máximos de las mediciones de campo eléctrico $E$ (V/m) realizadas sobre los 8 distritos de la ciudad de Alcalá en el año 2006 para la frecuencia de 900 MHz en exposición de público general. . . . .	13
2.1	Obtención de probabilidades de selección de individuos mediante la técnica de la ruleta para población de seis individuos, teniendo en cuenta que el algoritmo genético tiene como objetivo maximizar la función de evaluación. . . . .	28
2.2	Contenido de la <i>harmony memory</i> de la Figura 2.20 en forma matricial. Cada fila representa un acorde musical (solución del problema de optimización) y cada columna se corresponde con un instrumento musical (variable de decisión de la función objetivo). . . . .	45
4.1	Ejemplo de distribución de probabilidades de selección de progenitores por ruleta para una población de seis individuos a partir de sus valores de <i>fitness</i> . . . . .	60
6.1	Extracto del conjunto $\mathcal{P}$ utilizado en las simulaciones, formado por 509 puntos pertenecientes a la ciudad de Alcalá de Henares. . . . .	94

6.2	Tipos de estaciones base empleados para las simulaciones para una frecuencia de trabajo de 1800 MHz. . . . .	94
6.3	Resultados promedios obtenidos a partir de 60 simulaciones con el algoritmo evolutivo con diferentes valores de los coeficientes $k_1$ y $k_2$ en la función de evaluación $\mathcal{F}$ . El coeficiente $k_3$ se ha mantenido constante $k_3 = 1,75$ . . . . .	96
6.4	Mejores resultados obtenidos a partir de 60 simulaciones del algoritmo evolutivo con diferentes valores de los coeficientes $k_1$ y $k_2$ en la función de evaluación $\mathcal{F}$ . El coeficiente $k_3$ se ha mantenido constante $k_3 = 1,75$ . . . . .	96
6.5	Valores finales utilizados para los coeficientes $k_1$ , $k_2$ y $k_3$ de la función de evaluación $\mathcal{F}$ para los cinco algoritmos implementados. . . . .	96
6.6	Valores promedio para los parámetros que caracterizan las soluciones encontradas con el algoritmo PSO para diferentes tamaños de población. . . . .	100
6.7	Mejores individuos obtenidos con el algoritmo PSO variando el tamaño de la población. (**) Corresponden a las segundas mejores soluciones obtenidas en cada caso. . . . .	100
6.8	Mejores individuos obtenidos con cada uno de los algoritmos implementados. Los individuos aparecen ordenados de mayor a menor valor de <i>fitness</i> : algoritmo <i>Greedy</i> , PSO, HS, EA y CRO. (**) Corresponden a las segundas mejores soluciones obtenidas por sus correspondientes algoritmos. La representación gráfica para todos ellos puede verse en la Figura 6.6. . . . .	106
6.9	Valores promedio para los parámetros que caracterizan las soluciones encontradas con los algoritmos EA, <i>Greedy</i> , HS, PSO y CRO. . . . .	106

# Lista de Algoritmos

1	Pseudo-código del proceso de generación de la población inicial en el EA . . . . .	58
2	Pseudo-código del operador de cruce TBC en el EA . . . . .	62
3	Pseudo-código del operador de cruce DMC en el EA . . . . .	63
4	Pseudo-código para el operador de mutación en EA . . . . .	64
5	Pseudo-código del procedimiento de búsqueda local en un individuo con eliminación de estaciones base prescindibles . . . . .	69
6	Pseudo-código del procedimiento de búsqueda local en un individuo con eliminación y/o desplazamiento de estaciones base . . . . .	70
7	Pseudo-código general para el algoritmo <i>Greedy</i> . . . . .	72
8	Pseudo-código general para el algoritmo HS . . . . .	76
9	Pseudo-código general del algoritmo CRO . . . . .	86
10	Pseudo-código del proceso de generación de la colonia inicial en el algoritmo CRO . . . . .	87
11	Pseudo-código del proceso de eliminación de réplicas en el algoritmo CRO . . . . .	87
12	Pseudo-código del proceso de reproducción sexual en el algoritmo CRO . . . . .	88
13	Pseudo-código para el operador de mutación en el algoritmo CRO . . . . .	89
14	Pseudo-código del proceso de reproducción asexual en el algoritmo CRO . . . . .	90
15	Pseudo-código para el proceso de asentamiento en el arrecife en el algoritmo CRO . . . . .	91
16	Pseudo-código del proceso de depredación en el algoritmo CRO . . . . .	91
17	Pseudo-código para la generación de la población inicial en el GGA . . . . .	127
18	Pseudo-código del operador de cruce en el GGA . . . . .	128
19	Pseudo-código del operador de mutación en el GGA . . . . .	129
20	Pseudo-código de la operación de reparación del individuo en el GGA . . . . .	132
21	Pseudo-código del procedimiento de búsqueda local en un individuo en el GGA . . . . .	133

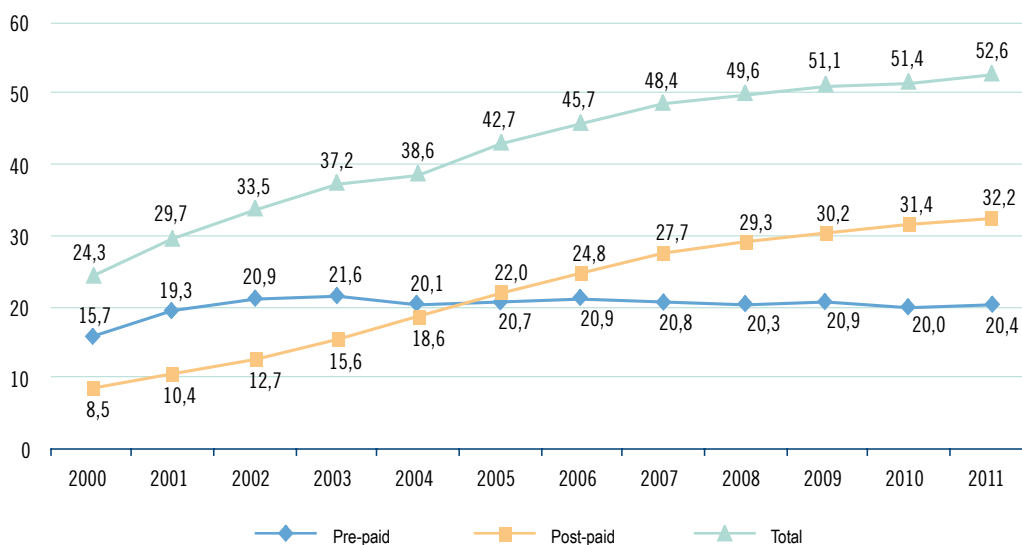


# Capítulo 1

## Introducción

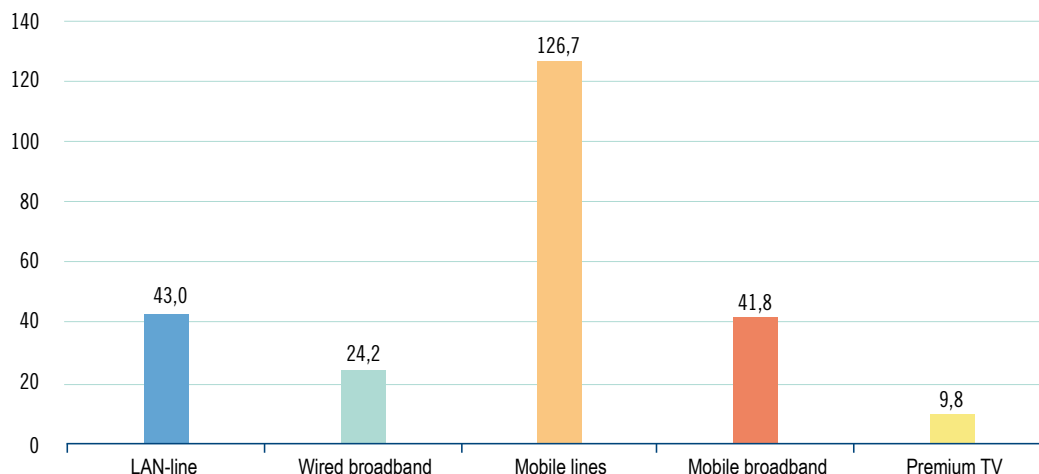
### 1.1 Introducción

En la sociedad actual el ciudadano medio no concibe la vida sin la existencia de los servicios de telecomunicación, especialmente los servicios de telefonía móvil y banda ancha. Las telecomunicaciones móviles han pasado a formar parte integral de nuestra vida. Hoy en día en las ciudades no es fácil encontrar una persona entre 14 y 66 años que no disponga de al menos, un terminal móvil. Como se relaciona en [24], la Figura 1.1 ofrece el parque de líneas personales de voz (donde líneas móviles vinculadas a *datacards* están excluidas) en España durante los años 2000-2011. En el año 2010 superaron los 51,4 millones de líneas. En el año 2011, las líneas personales de voz superaron los 52,6 millones. El crecimiento de los 1,2 millones de líneas entre los años 2010 y 2011, se observó tanto en el segmento de *post-pago*, con un incremento del 2,5%, como en el segmento de *pre-pago*, el cual experimentó un aumento interanual del 2%. Las líneas asociadas a *datacards* o máquinas son líneas vinculadas a servicios de telemetría o telecontrol. Según se recoge en [24], el parque de líneas asociadas a *datacards* alcanzó en 2011 los 2,5 millones, lo que supuso un aumento del 18,1% respecto al ejercicio anterior.

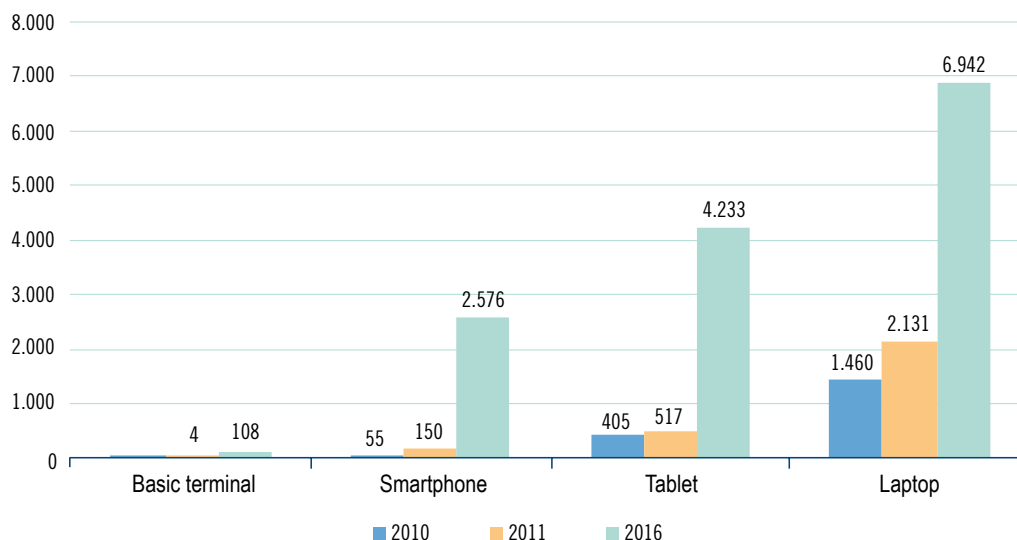


**Figura 1.1:** Evolución del número de líneas móviles, en millones de líneas, en España durante los años 2000-2011. Fuente: Comisión del Mercado de las Telecomunicaciones (CMT).

El mercado de las telecomunicaciones se encuentra en continuo crecimiento. La Figura 1.2 representa gráficamente el grado de penetración de los principales servicios en España a finales del año 2011. Como puede verse en la gráfica, el número de líneas destinadas al servicio de voz está entrando en la zona de saturación alcanzando las 126 líneas por cada 100 habitantes. Sin embargo, los servicios de datos de banda ancha móvil están tomando el relevo [23], con un crecimiento del 171% entre los años 2009 y 2010. La Figura 1.3 muestra la previsión realizada por [24] de tráfico medio estimado de banda ancha móvil para el año 2016 con los dispositivos más comunes.



**Figura 1.2:** Grado de penetración de los principales servicios de telecomunicación en España a fecha el mes de diciembre de 2011, en número de líneas por cada 100 habitantes. Fuente: *CMT*.



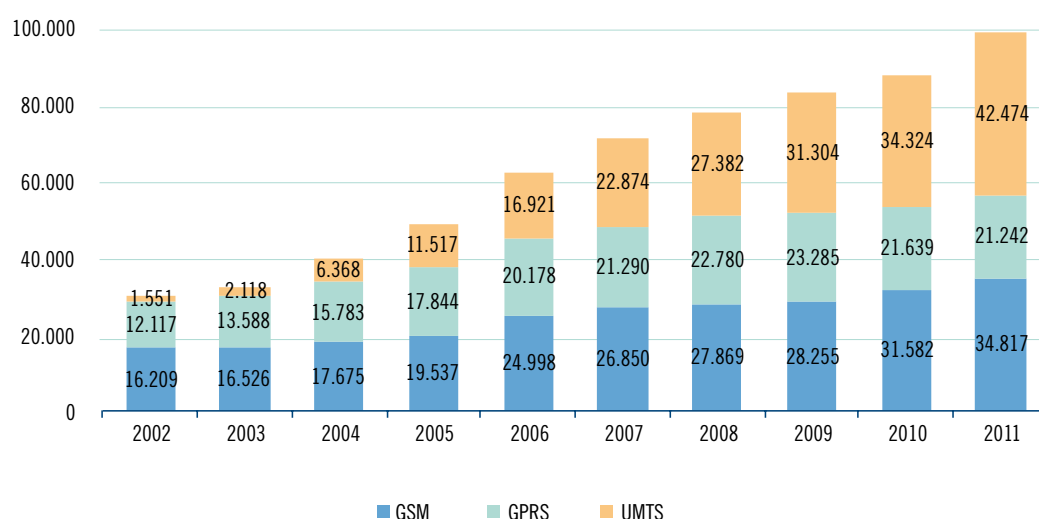
**Figura 1.3:** Tráfico medio estimado de banda ancha móvil por tipo de dispositivo en España para 2016 (MB/mes). Fuente: *CMT*.

Internet se extiende sin receso en todos los campos, influyendo cada vez más en ámbitos como educación, ciencia e investigación, sanidad, industria y hasta en las políticas interior y exterior



de cada país. Incluso la supervivencia y el auxilio de remotos pueblos con escasa población dependen del acceso que éstos tengan a las redes de telecomunicación. Es más, ha cambiado la forma en la que los individuos se relacionan, especialmente entre los más jóvenes, con el uso de las conocidas redes sociales que permiten una comunicación permanente con un grupo amplio de personas.

Según aumenta la diversidad y mejora de los servicios de telecomunicación, se incrementa también la necesidad de ampliar las infraestructuras que a su vez satisfagan la correspondiente demanda. En el caso de la telefonía móvil, consiste en un mayor despliegue de estaciones base o BTS (*Base Transceiver Station*) y nodos B instalados; para las tecnologías de segunda generación o GSM (*Global System for Mobile Communications*) y de tercera generación o UMTS (*Universal Mobile Telephone System*), respectivamente. La Figura 1.4 muestra la evolución del número de estaciones base por tecnología entre los años 2002 y 2011 [24]. Las estaciones base son los elementos de la red de acceso más numerosos en la red, junto con los terminales móviles o estaciones móviles MS (*Mobile Station*) adquiridos por los usuarios.



**Figura 1.4:** Evolución del número de estaciones base implantadas según la tecnologías GSM, GPRS y UMTS en España durante la década 2002-2011. Fuente: CMT.

Es de conocimiento general el hecho de que la sociedad actual no acepta prescindir de los servicios de telecomunicación en la vida cotidiana, sin embargo sí desearía evitar algunas de las implicaciones que la instalación de equipamiento necesario conlleva. Estas implicaciones son fundamentalmente tres: la dedicación de terrenos para las instalaciones radioeléctricas, la alteración visual del entorno paisajístico debido a estas instalaciones y la radiación electromagnética emitida por las mismas. La asignación de superficies para la colocación de equipos es sobrellevada por la sociedad de forma aceptable. El avance de la tecnología reduce progresivamente el tamaño del equipamiento, si bien es cierto que las necesidades de nuevos equipos crecen de manera constante. De cualquier modo, el espacio dedicado a estos menesteres se encuentra acotado.

El impacto visual sobre el paisaje cobra importancia en áreas no urbanas. En las ciudades este hecho es asumido por completo. La predisposición de los habitantes hacia el progreso tecnológico en las zonas rurales es menos positiva que en las ciudades. No obstante, el número de usuarios en zonas no urbanas es muy reducido con respecto al total de usuarios.

La última inquietud por parte de los usuarios frente al desarrollo de las telecomunicaciones, y desde luego la más importante, es la radiación que emiten las estaciones base y los posibles efectos adversos para la salud. La radiación electromagnética no es más que una propagación de energía de esta naturaleza a través del aire desde un punto emisor o fuente. Los equipos emisores de radiación electromagnética son tanto las estaciones base (*BTSs* y nodos B) como las propias estaciones móviles que portan los usuarios. A las frecuencias designadas a los servicios de telefonía móvil (baja frecuencia o microondas), la energía de un fotón no es suficiente para romper uniones atómicas. Es por ello que a este tipo de radiación se dice que es no-ionizante de baja frecuencia, esto quiere decir, que no posee energía suficiente para ionizar la materia o producir radiactividad en el cuerpo. Para más información acerca de los tipos de radiación electromagnética ver Anexo A. Sin embargo este tipo de radiofrecuencia es capaz de ceder energía suficiente al cuerpo radiado como para producir efectos térmicos. La cuantía de estos efectos dependen de diversos parámetros como la frecuencia, la orientación, el tamaño y las propiedades del cuerpo expuesto a dicha radiación.

Un porcentaje destacado de usuarios sienten rechazo hacia las antenas de las estaciones base, aunque de manera poco reflexiva se sienten cómodos y seguros llevando consigo sus teléfonos móviles, que emiten éstos incluso radiación más intensa. La población general percibe que el uso de un terminal móvil es un acto voluntario cuya utilización aporta intrínsecamente beneficios, mientras que la presencia cercana de una estación base es algo que el usuario no ha decidido ni individual ni voluntariamente. De forma que los beneficios de estos equipamientos no son percibidos fácilmente por la población. Además existe cierto grado desconocimiento general sobre las razones para la ubicación de las antenas. Con el despliegue de las redes y servicios móviles se ha extendido un sentimiento de desasosiego y miedo a la radiación de las antenas de telefonía móvil (ver artículo de la Asociación Española Contra el Cáncer en [4]).

Hay que hacer notar que esta percepción de peligro no existía años anteriores en convivencia con otras fuentes de radiación como las antenas de los sistemas buscapersonas y sistemas de comunicación utilizados por la policía, bomberos y cuerpos de emergencia. Estos operan a frecuencias similares y con niveles de potencia cercanas a los de las estaciones bases. En muchas áreas urbanas las antenas de transmisión de radiodifusión sonora y televisión normalmente operan a niveles de radiofrecuencia mayores que los de las estaciones bases de telefonía móvil. Los teléfonos móviles emiten potencias máximas entre 0,2 a 0,6 W. Otros tipos de transmisores portátiles, como los *walkie talkie*, pueden emitir 10 W o superarlos.

La intensidad del campo de radiofrecuencia desciende rápidamente con la distancia a la fuente de emisión. Esto significa que una persona que utiliza su teléfono móvil se somete en general a una exposición de energía electromagnética considerablemente mayor que otra persona viviendo cerca de una estación base (ver artículo de la Organización Mundial de la Salud en [77]). Además, las antenas instaladas sobre edificios en zonas urbanas orientan su energía al exterior y no irradian cantidades significativas de energía hacia la base de la antena. De esta manera, los niveles de energía dentro o en los lados del edificio son normalmente bajos.

La preocupación por la radiación de las antenas, alimentada en muchos casos por el desconocimiento y la falta de información veraz, se traduce en una animadversión a la presencia de las estaciones base y la completa oposición a la instalación de nuevas estaciones. Este conflicto ya ha afectado a multitud de ciudades, tanto en España (como Alcalá de Henares, Torrejón de Ardoz, Leganés, Murcia, Zaragoza y Basauri entre otros) como a nivel internacional.

## 1.2 Radiación electromagnética y límites de exposición

Las telecomunicaciones móviles en España tienen asignadas las frecuencias de trabajo: 900 Megahercios (MHz), 1800 MHz, y 2100 MHz, ver Anexo A. La frecuencia de 900 MHz se dedica fundamentalmente a GSM y una pequeña parte para UMTS [24] <sup>1</sup>. La frecuencia 1800 MHz se utiliza fundamentalmente para servicios de GPRS y la de 2100 MHz se emplea para UMTS.

Como ya hemos mencionado en el apartado anterior, la radiación electromagnética producida por los sistemas de telecomunicación móviles es de tipo no-ionizante dado que no posee energía suficiente para ionizar la materia. Cuando un ser vivo se encuentra en el área de influencia de un campo de radiofrecuencia, parte de la energía radiada es absorbida por el cuerpo, penetrando en los tejidos a una profundidad que varía en función de la frecuencia. La energía absorbida produce un calentamiento de los tejidos vivos. Los mecanismos naturales de termo-regulación disipan este calor manteniendo la temperatura corporal dentro de unos límites adecuados biológicamente.

Existen numerosos organismos y comités científicos, tanto a nivel nacional como internacional, que estudian las reacciones del cuerpo humano a la exposición radioeléctrica y documentan los posibles efectos en la salud pública [22]. Estas organizaciones establecen valores de rangos de radiación dentro de los cuales no se reconoce peligro para la salud. A continuación ofrecemos una relación de los organismos con prestigio reconocido que realizan este tipo de estudios de investigación:

- Organización Mundial de la Salud (OMS) <sup>2</sup>: Es una agencia especial de las Naciones Unidas formada por 191 estados. La OMS colabora activamente con otros organismos internacionales entre los que destaca la ICNIRP, la UIT y la Comisión Europea.
- *International Commission for Non-Ionizing Radiation Protection* (ICNIRP) <sup>3</sup>: la Comisión Internacional para la Protección contra las Radiaciones No Ionizantes es una comisión científica independiente. Fue creada por la Asociación Internacional de Protección Radiológica en Alemania en el año 1974.
- *The Royal Society of Canada* <sup>4</sup>: es el primer organismo fundado para estas actividades, creado en el año 1882.
- SCENIHR <sup>5</sup>: es el Comité Científico de la Unión Europea sobre Nuevos Riesgos Emergentes Identificados en la Salud. Sustituye al antiguo CSTEE (*Scientific Committee on Toxicity, Ecotoxicity and the Environment*).
- *Australian Radiation Protection and Nuclear Safety Agency* (ARPANSA) <sup>6</sup>: es una agencia gubernamental australiana.
- *Committee on Man and Radiation* (COMAR) <sup>7</sup>: es el Comité Técnico de la Sociedad de Ingeniería en Medicina y Biología del IEEE.

---

<sup>1</sup>El Informe Económico Sectorial 2011 de la CMT ([24]) cita textualmente: “la Directiva GSM 87/372/CE modificada y la Decisión 2009/766/CE a nivel europeo introdujeron el principio de neutralidad tecnológica y de servicios. Una de las consecuencias de la aplicación de este principio es que se pueden utilizar las bandas de 900 MHz y de 1.800 MHz, no solo para sistemas basados en tecnología GSM, sino también para otros sistemas que posibiliten mayores velocidades de transmisión de datos, como los basados en la tecnología UMTS, LTE u otras compatibles. Esta medida puede ayudar a impulsar el despliegue de sistemas de acceso a banda ancha móvil, especialmente en zonas rurales”.

<sup>2</sup><http://www.who.int/en/>

<sup>3</sup><http://www.icnirp.de/>

<sup>4</sup><http://rsc-src.ca/>

<sup>5</sup><http://ec.europa.eu/>

<sup>6</sup><http://www.arpansa.gov.au/>

<sup>7</sup><http://ewh.ieee.org/soc/embs/comar/>

- *National Radiological Protection Board* (NRPB) <sup>1</sup>: el Consejo Nacional para la Protección contra la Radiación es una organización inglesa que desde abril de 2013 forma parte de la sanidad pública de Inglaterra.

Otros organismos de menor prestigio que redactan recomendaciones son: *Australian Communications and Media Authority* (ACMA), Asociación Española de Normalización y Certificación (AENOR), *American National Standard Institute* (ANSI), *European Telecommunications Standard Institute* (ETSI), *International Electrotechnical Commission* (IEC) y *Institute of Electrical and Electronics Engineers* (IEEE), entre otros. Solo dos organizaciones tienen capacidad de emitir normativas de obligado cumplimiento:

- *European Committee for Electrotechnical Standardization* (CENELEC): se constituyó en 1973 y elabora estándares que se aplican en toda Europa, se desarrollan de forma conjunta con la *International Electrotechnical Commission* (IEC).
- *Federal Communications Commission* (FCC): es una agencia gubernamental independiente de EEUU. Se fundó en 1934, y se encarga de regular las comunicaciones por radio, televisión, cable y satélite en el país.

En España, la función regulativa la realiza el Ministerio de Ciencia y Tecnología (actualmente el Ministerio de Industria, Turismo y Comercio) de forma conjunta con el Ministerio de Sanidad y Consumo. Según la Ley 11/1998 General de Telecomunicaciones (actualizada en la Ley 32/2003), la inspección de las redes de telecomunicaciones corresponde al Ministerio de Industria, que a su vez adjudica a la Secretaría de Estado de Telecomunicaciones y para la Sociedad de la Información la competencia de gestión del dominio público radioeléctrico y comprobación técnica de emisiones radioeléctricas.

Por otro lado, la Ley 14/1986 General de Sanidad atribuye las competencias sobre formas de energía que puedan suponer un riesgo para la salud al Ministerio de Sanidad y Consumo. El Real Decreto 1450/2000 asume los criterios de protección sanitaria frente a los campos electromagnéticos establecidos en la Recomendación del Consejo de Ministros de Sanidad de la Unión Europea (12 de julio de 1999) basada en los criterios de la ICNIRP. El Boletín Oficial del Estado (BOE) tiene varias publicaciones posteriores al Real Decreto 1450/2000 en materia de protección de la población frente a emisiones radioeléctricas <sup>2 3 4 5</sup>.

Las magnitudes físicas utilizadas en recomendaciones y normativas en esta materia son fundamentalmente: la densidad de corriente ( $J$ ), la Tasa Específica de Absorción de energía o *Specific energy Absorption Rate* ( $SAR$ ) y la densidad de potencia emitida ( $S$ ). Otras unidades también empleadas son la intensidad de campo eléctrico ( $E$  medida en V/m), la intensidad de campo

<sup>1</sup><http://www.hpa.org.uk/radiation/>

<sup>2</sup>Real Decreto 1066/2001, de 28 de septiembre, por el que se aprueba el Reglamento que establece condiciones de protección del dominio público radioeléctrico, restricciones a las emisiones radioeléctricas y medidas de protección sanitaria frente a emisiones radioeléctricas. BOE núm. 234, de 29 de septiembre de 2001, BOE núm. 257, de 26 de octubre de 2001 y BOE núm. 93, de 18 de abril de 2002.

<sup>3</sup>Orden CTE/23/2002, de 11 de enero, por la que se establecen condiciones para la presentación de determinados estudios y certificaciones por operadores de servicios de radiocomunicaciones. BOE núm. 11, de 12 de enero de 2002 y BOE núm. 117, de 16 de mayo de 2002.

<sup>4</sup>Disposición final cuarta del Real Decreto 424/2005, de 15 de abril, por el que se aprueba el Reglamento sobre las condiciones para la prestación de servicios de comunicaciones electrónicas, el servicio universal y la protección de los usuarios, que modifica el apartado 1 del artículo 8 y el apartado 3 del artículo 9 del Real Decreto 1066/2001, de 28 de septiembre.

<sup>5</sup>Resolución de 22 de mayo de 2007, de la Secretaría de Estado de Telecomunicaciones y para la Sociedad de la Información, por la que se sustituye la inspección previa al uso del dominio público radioeléctrico de determinadas estaciones radioeléctricas por una certificación expedida por técnico competente.

magnético ( $H$ , medido en A/m), la densidad de flujo magnético ( $B$ , medido en teslas (T)), las corrientes que fluyen a través de las extremidades ( $I_L$ ) y las corrientes de contacto ( $I_C$ ). La distribución de la densidad de corriente inducida varía inversamente con la sección transversal del cuerpo y puede ser relativamente alta en el cuello y tobillos. A continuación comentamos brevemente algunas de ellas.

- $SAR$  (unidades de W/kg): es la medida de la energía absorbida por unidad de masa de tejido corporal expuesto. La  $SAR$  es proporcional al cuadrado de la intensidad de campo eléctrico interno. A frecuencias entre 300 MHz y varios GHz se produce una absorción de energía no-uniforme significativamente local. Esta magnitud es la magnitud de referencia internacionalmente para relacionar los efectos térmicos causados por radiofrecuencia.
- Densidad de potencia emitida ( $S$  medida en  $W/m^2$ ): es la potencia por unidad de área normal a la dirección de propagación. Solo la densidad de potencia en el aire fuera del cuerpo puede ser rápidamente medida en individuos en exposición a radiofrecuencia.
- La densidad de corriente  $J$  y la intensidad de campo eléctrico  $E$  se relacionan por la ley de Ohm  $J = \sigma E$ , donde  $\sigma$  es la conductividad eléctrica del medio medida en siemens por metro (S/m).
- La densidad de flujo magnético  $B$  y la intensidad de campo magnético  $H$  se relacionan mediante la ecuación  $B = \mu \times H$ , donde  $\mu$  es la permeabilidad magnética del medio, con unidades henrios por metro (H/m). El valor de la permeabilidad en el vacío es  $\mu_0 = 4\pi 10^{-7}$  (H/m), dicho valor se utiliza como aproximación para el aire y también para material biológico.

En todas las recomendaciones y disposiciones legales se consideran dos tipos bien diferenciados de exposición a radiación según el tipo de población involucrada: exposición controlada y no controlada. Cada una tiene a su vez restricciones básicas:

- Exposición controlada u ocupacional: es el caso de adultos desarrollando una actividad profesional en zonas de radiación conocida y controlada. Estas personas son conocedoras de su exposición y entrenadas para tomar las precauciones de seguridad necesarias.
- Exposición no controlada o de público general: comprende el conjunto de individuos de todas las edades y estados de salud en general sin conocimiento de la radiación que reciben. Como no son conscientes de su exposición, no pueden ejercer control alguno sobre la misma. Por esta razón las restricciones son de carácter más estricto para la exposición de público general que para la de población ocupacional.

Se definen un conjunto de restricciones básicas y sus correspondientes niveles de referencia. Las primeras definen límites a la exposición de radiación basados en los criterios de las comisiones FCC y ICNIRP. Las restricciones básicas se expresan en términos de densidad de corriente en cabeza y tronco y de la  $SAR$  medida en promedio en el cuerpo y de forma localizada en cabeza/tronco y extremidades.

Los niveles de referencia son producto de modelos matemáticos y tienen como propósito determinar de forma práctica la probabilidad de que dichas restricciones básicas sean excedidas. Los niveles de referencia se expresan en términos de la intensidad de campo eléctrico, intensidad de campo magnético y la densidad de potencia. El respeto de todos los niveles de referencia asegura el cumplimiento de las restricciones básicas. Si en algún caso el valor de un parámetro

medido sobrepasa el nivel de referencia, no implica que se exceda la restricción básica correspondiente. Simplemente significa que es preciso evaluar explícitamente el cumplimiento de dicha restricción y determinar si son necesarias medidas de protección adicionales.

La distribución de la *SAR* promedio puede ser calculada y estimada de forma experimental. La medida de la *SAR* depende de: parámetros de los campos incidentes (como frecuencia, intensidad, polarización), de las propiedades físicas del tejido expuesto (tamaño, geometría interna y externa, propiedades dieléctricas), de los efectos de la tierra eléctrica y de los efectos de reflexión de otros objetos en el campo cercano del cuerpo expuesto [16]. La evidencia experimental indica que la exposición a campos electromagnéticos de seres humanos en reposo durante aproximadamente 30 minutos produce un *SAR* en todo el cuerpo entre 1 y 4 W/kg, incrementándose la temperatura corporal en 1°C como máximo. Se sabe que en los tejidos sensibles se produce daño irreversible a partir de un valor de *SAR* de 4 W/kg o superior, promediados en todo el cuerpo y bajo condiciones ambientales normales. Valores superiores de *SAR* de 4 W/kg son potencialmente capaces de producir efectos nocivos para la salud como:

- Incrementos térmicos de 1 °C durante intervalos breves puede reducir la habilidad para el desarrollo de ciertas tareas físicas o intelectuales.
- Incrementos de 2-3 °C durante intervalos prolongados pueden producir infertilidad temporal en varones, daño fetal o inducción de cataratas.

Las comisiones reguladoras ICNIRP y FCC consideran únicamente los efectos térmicos sobre la población. No existen estudios concluyentes ni evidencias consistentes sobre posibles efectos no térmicos derivados de la radiofrecuencia en seres vivos [57].

La Tabla 1.1 muestra los niveles máximos de *SAR* admitidos por estas comisiones en exposición controlada y sobre público general. Ambas comisiones coinciden en el valor numérico de la restricción básica para la *SAR* promedio en todo el cuerpo en exposición ocupacional (0,4 W/kg). Comparando este dato con el umbral experimental de 4 W/kg para reconocer daño irreversible, se considera un margen de seguridad suficiente ante condiciones variables tales como temperatura ambiente, humedad o actividad física del sujeto. Respecto a las restricciones en público general, hay un factor de multiplicación de 5 para todos los casos. Con este factor marca un amplio margen para garantizar la protección frente a una diversidad de situaciones e individuos (primero los niños, ancianos y enfermos tienen mayor sensibilidad térmica que el resto de individuos; segundo ciertos medicamentos modifican significativamente la tolerancia térmica; y tercero ciertas drogas y el alcohol alteran la capacidad termo-reguladora del cuerpo). Para una explicación más detallada ver [85].

**Tabla 1.1:** Restricciones básicas de *SAR* (W/kg) según la ICNIRP y la FCC en exposiciones a campos electromagnéticos en el rango de frecuencias [100 KHz - 6 GHz] con un tiempo medio de exposición de 6 minutos. Las medidas de *SAR* en zonas parciales del cuerpo se refieren a 10 gr de tejido contiguo.

	Ocupacional			Público general		
	Cuerpo gal.	Cabeza/tronco	Extremidades	Cuerpo gal.	Cabeza/tronco	Extremidades
ICNIRP	0.4	10	20	0.08	2	4
FCC	0.4	8	20	0.08	1.6	4

Las siguientes tablas permiten comparar los niveles máximos establecidos por diferentes organismos para las magnitudes de intensidad de campo eléctrico *E*, intensidad de campo magnético

$H$  y densidad de potencia equivalente  $S$  en entornos tanto controlados (exposición ocupacional) como no controlados (exposición público general). Consideramos de forma particular las distintas frecuencias de telefonía móvil: 900 MHz en la Tabla 1.2, 1800 MHz en la Tabla 1.3 y 2100 MHz en la Tabla 1.4. Hemos querido destacar en color rojo la comisiones con capacidad reguladora en Europa (ICNIRP) y en EEUU (FCC). Esta última solo publica valores máximos permitidos para la densidad de potencia equivalente  $S$ , no para  $E$  ni  $H$ .

**Tabla 1.2:** Niveles de referencia: Valores máximos de  $E$ ,  $H$  y  $S$  establecidos por los diferentes organismos para la frecuencia de 900 MHz en exposición controlada (ocupacional) y no controlada (público general). Nota: los organismos en rojo indican márgenes de obligado cumplimiento en Europa y EEUU.

	Ocupacional			Público general		
	$E(V/m)$	$H(A/m)$	$S(W/m^2)$	$E(V/m)$	$H(A/m)$	$S(W/m^2)$
ARPANSA	92.1	0.244	22.5	41.1	0.109	4.5
ANSI/IEEE	-	-	30	-	-	4.5
M. Canadá	106.2	0.282	30	47.55	0.126	6
ICNIRP	90	0.24	22.5	41.25	0.111	4.5
FCC	-	-	30	-	-	6

**Tabla 1.3:** Niveles de referencia: Valores máximos de  $E$ ,  $H$  y  $S$  para la frecuencia de 1800 MHz, establecidos por los diferentes organismos en exposición controlada y no controlada. Nota: los organismos en rojo indican márgenes de obligado cumplimiento en Europa y EEUU.

	Ocupacional			Público general		
	$E(V/m)$	$H(A/m)$	$S(W/m^2)$	$E(V/m)$	$H(A/m)$	$S(W/m^2)$
ARPANSA	130.249	0.345	45	58.124	0.154	9
ANSI/IEEE	-	-	60	-	-	9
M. Canadá	150.189	0.399	60	67.246	0.178	12
ICNIRP	127.279	0.339	45	58.336	0.157	9
FCC	-	-	60	-	-	12

Se observa en las tablas que para las tres frecuencias, la densidad de potencia equivalente en el caso de exposición sobre público general es cinco veces inferior a los valores límites de la misma magnitud en exposición ocupacional. Los valores máximos están delimitados con amplios márgenes de seguridad para preservar la protección sanitaria de los ciudadanos. En cuanto a los niveles de referencia marcados por la ICNIRP para los campos eléctrico  $E$  y magnético  $H$  en exposición no controlada son menores en aproximadamente un factor de  $5^{1/2} \approx 2,2$  con respecto a los niveles de exposición controlada, siendo 5 el factor que relaciona los valores de referencia para la densidad de potencia en los dos tipos de exposición.

Finalmente la Tabla 1.5 muestra los valores de referencia de densidad de potencia equivalente  $S$  en entornos no controlado o exposición de público general adoptados en diferentes países. Destacan China y Rusia como las más restrictivas, mientras que EEUU es el menos exigente con los límites de nivel de radiación. Bélgica se convierte en una excepción de la Unión Europea como país muy sensibilizado en estos temas.

**Tabla 1.4:** Niveles de referencia: Valores máximos de E, H y S establecidos por los diferentes organismos para la frecuencia de 2100 MHz en exposición controlada y no controlada. Nota: los organismos en rojo indican márgenes de obligado cumplimiento en Europa y EEUU.

	Ocupacional			Público general		
	E(V/m)	H (A/m)	S(W/m <sup>2</sup> )	E(V/m)	H (A/m)	S(W/m <sup>2</sup> )
ARPANSA	137	0.364	50	61.4	0.163	10
ANSI/IEEE	-	-	70	-	-	10
M. Canadá	137	0.364	50	61.4	0.163	10
ICNIRP	137	0.36	50	61	0.16	10
FCC	-	-	50	-	-	10

**Tabla 1.5:** Valores máximos de  $S$  establecidos por los diferentes organismos para la frecuencia de 2100 MHz en exposición no controlada. (\*) agrupa la mayoría de los países miembros de la Unión Europea: Alemania, Austria, Dinamarca, Eslovaquia, Eslovenia, España, Estonia, Finlandia, Francia, Grecia, Holanda, Australia y Nueva Zelanda.

	900 MHz	1800 MHz	2100 MHz
Unión Europea*	4,5	9	10
Bélgica	1,125	2,25	2,5
EEUU	6	12	14
Canadá	6	12	10
Japón	6	12	10
China	0,4	0,4	0,4
Rusia	0,1 si $t_{exp} = 8 h$	1 si $t_{exp} \leq 2 h$	10 si $t_{exp} \leq 20'$

### 1.3 Estudio de las emisiones radioeléctricas en la ciudad de Alcalá de Henares

Durante el año 2006, investigadores del Departamento de Teoría de la Señal y Comunicaciones de la universidad realizaron un proyecto de investigación conjunto con el Excmo. Ayuntamiento de Alcalá de Henares. Este proyecto se titula: “Estudio de las emisiones radioeléctricas en las bandas de frecuencias asignadas a los servicios de telefonía móvil, en la ciudad de Alcalá de Henares”. El grupo investigador que lo llevó a cabo estuvo integrado por once investigadores del departamento con distintos perfiles de formación y especialización.

El objetivo principal del proyecto consistió en la obtención de medidas de radiación electromagnética de tipo no-ionizante sobre la ciudad de Alcalá para un determinado rango de frecuencias, generando de esta manera un mapa radioeléctrico de la ciudad. Para ello se realizaron las correspondientes mediciones de niveles de radiación electromagnética en todo el área de la ciudad de Alcalá de Henares, tomando especial interés en las frecuencias asignadas a los servicios de telefonía móvil (900-2100 MHz). Un segundo objetivo comprendió la comprobación de que las medidas obtenidas experimentalmente no superaban en ningún caso los niveles de referencia marcados por la ICNIRP e indicados en las Tablas 1.2, 1.3 y 1.4 que garantizan la protección sanitaria de la población frente a este tipo de emisiones radioeléctricas. Este proyecto tenía como última misión informar de manera detallada y fiable a los habitantes de Alcalá de Henares



acerca del grado de radiación en la ciudad, para apagar la inquietud y reducir el desconocimiento general entre la población sobre los efectos de estaciones base de telefonía móvil.

La realización de este estudio y medición de emisiones en la banda de frecuencias de telefonía móvil tiene especial importancia para el desarrollo de nuestro trabajo de tesis puesto que los distintos algoritmos implementados han tomando como fuente de datos el conjunto de mediciones realizadas de manera experimental en Alcalá de Henares.

Presentamos aquí brevemente los puntos principales de este proyecto, el protocolo de medida realizado y el análisis de los resultados obtenidos. El protocolo llevado a cabo primero definía el entorno de actuación para después proceder a las medidas en banda ancha y medidas en banda estrecha, si estas últimas eran necesarias.

Se identificaron un total de 50 puntos fuente de emisión radioeléctrica. Estos fueron localizados a través de la página de Internet del Ministerio de Industria, Turismo y Comercio <sup>1</sup> y con la colaboración de la Federación Comarcal de Asociaciones de Vecinos de Alcalá de Henares. También se clasificaron los diferentes tipos de servicios de telecomunicación contribuyentes a la radiación. Se estableció la cantidad y la posición del conjunto de puntos donde realizar las medidas de campo eléctrico. El protocolo definió el periodo de realización de mediciones así como la franja horaria en la que llevarlas a cabo. Todas las medidas se realizaron entre los meses de abril a septiembre del año 2006 y siempre en días laborables y en hora punta de utilización de servicios de telefonía. Las franjas horarias de máxima utilización corresponden entre las 10 horas de la mañana y 12 horas del mediodía y en entre las 16 horas y las 20 horas de la tarde.

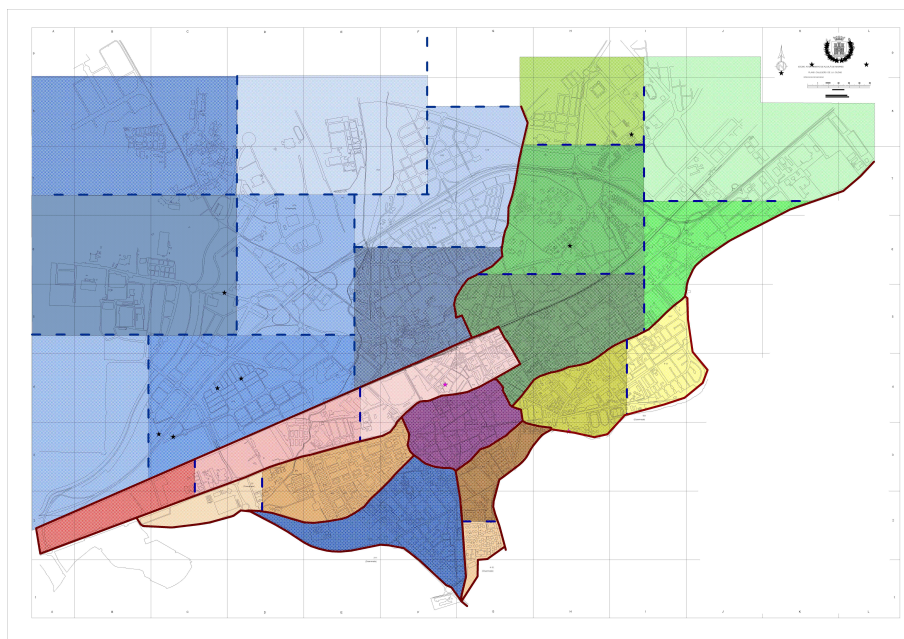
En cuanto a la ubicación de los puntos de medición, se partió de una formación reticular uniforme sobre la ciudad compuesta por celdas cuadradas de 200 m de lado. Sin embargo la distribución uniforme no era la mejor opción puesto que ni la población ni los denominados “puntos sensibles” se encuentran repartidos de forma uniforme en la ciudad. Los “puntos sensibles” son zonas en las que se encuentran individuos con una sensibilidad superior a la media frente a la radiación. Es el caso de guarderías, centros de enseñanza, centros de salud, hospitales, parques públicos, centros y residencias de tercera edad. Por otra parte una distribución teórica bajo una cuadrícula podría suponer realizar algunas mediciones en zonas de acceso difícil o inviable, por ejemplo en el interior de un lago o el el centro de una vía rápida de vehículos. En conclusión, la distribución de puntos de medida se modificó manualmente prescindiendo de algunos puntos que se mostraban innecesarios y añadiendo otros nuevos para determinar finalmente la ubicación y cantidad de puntos de medida a realizar de acuerdo a las características de: tamaño de la ciudad, población y ubicación de las distintas fuentes de emisión radioeléctrica de tipo no-ionizante, así como situación de los denominados puntos sensibles en la ciudad Alcalá. El número final de puntos de medida se fijó en 509 puntos.

La ciudad de Alcalá de Henares está compuesta por un conjunto de distritos como puede verse en la Figura 1.5. Esta imagen ha sido cedida directamente por los investigadores del proyecto y no la hemos querido modificar, a excepción de reducir su tamaño original. Los distritos quedan definidos en la figura por una línea continua de color granate. Algunos de ellos tienen suficiente extensión como para desglosarse a su vez en varias partes que aparecen formadas como si de una rejilla se tratara. Sin embargo esta información no afecta al trabajo de tesis y por lo tanto no queremos extendernos en ello.

La Figura 1.6 también está tomada de los planos del proyecto y se encuentra reducida en tamaño. En ella se muestra la ciudad de Alcalá sobre la que pueden apreciarse los ocho distritos, los 50 puntos radiantes y los 509 puntos en los que se realizaron medidas de campo electromagnético. Debido a la reducción de tamaño de la imagen prestada del proyecto de investigación,

---

<sup>1</sup>Ministerio de Industria: servicio de información de instalaciones radioeléctricas y niveles de exposición ([www.mitic.es/nivelesexposicion](http://www.mitic.es/nivelesexposicion))

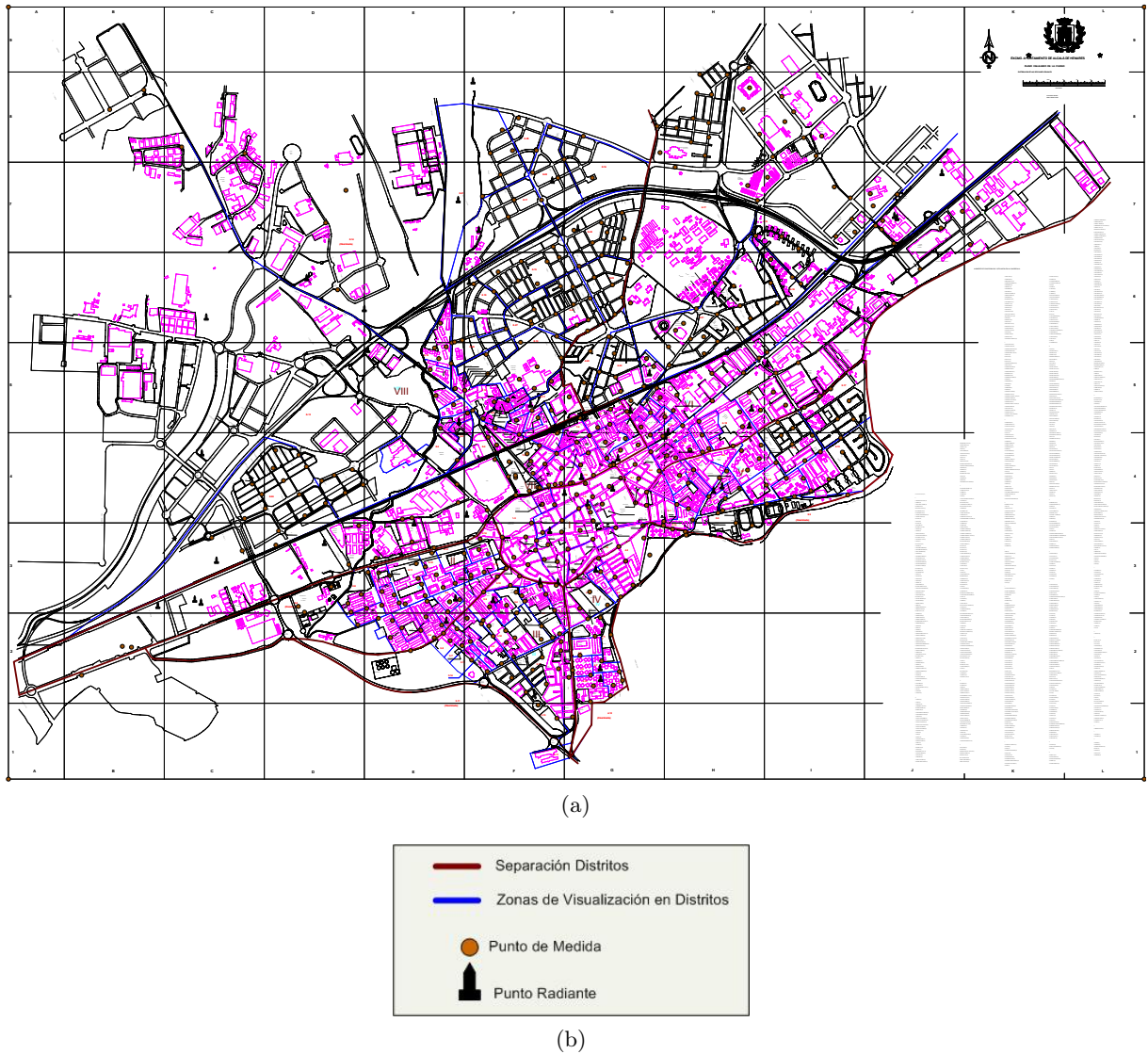


**Figura 1.5:** Agrupación de los ocho distritos que componen la ciudad de Alcalá de Henares. Cada uno de ellos queda circunscrito por una línea de color granate.

resulta complicado la lectura de la leyenda de la Figura 1.6(a) donde se enumeran los puntos de medición. Como ya se ha explicado, dado que esta información no es relevante para este trabajo de tesis, no nos ha parecido necesario ampliar las figuras. Nosotros hemos trabajado con hojas de cálculo cedidas por el proyecto conteniendo los datos de las mediciones al completo.

Las mediciones en banda ancha se realizaron para 509 los puntos con la ayuda de un medidor de radiación de banda ancha y una sonda isotrópica para la medida de la intensidad de campo eléctrico. La duración media de las mediciones fue de seis minutos. Las medidas en banda estrecha tienen bastante mayor precisión que las de banda ancha. Se emplean cuando los niveles recogidos en banda ancha exceden los niveles de decisión. Los niveles de decisión se sitúan 6 dB por debajo de los niveles de referencia establecidos por la ICNIRP. Esta diferencia supone que el nivel de decisión para el campo eléctrico, en unidades naturales, es la mitad del nivel de referencia correspondiente. El protocolo de medida determina que cuando una medición de banda ancha supere el nivel de decisión, entonces ha de repetirse dicha medición con aparatos más precisos para obtener con precisión los niveles de radiación (medición en banda estrecha). En ninguno de los puntos se superaron los niveles de decisión, aunque sí se realizaron medidas de banda estrecha para conocer el nivel de radiación para cada frecuencia en particular. El material utilizado para banda estrecha fue un analizador de espectros y antenas que permitieran la medida de campo en polarización vertical y horizontal. La Tabla 1.6 muestra los valores máximos de las mediciones de campo eléctrico realizadas para la frecuencia de 900 MHz en exposición de público general. Como puede observarse, estos valores son muy inferiores al nivel de referencia establecido por la ICNIRP en la Tabla 1.2, con un valor de campo  $E$  de 41,25 V/m.

Finalmente recogemos también del proyecto y con el tamaño reducido, la Figura 1.7 que representa el mapa radioeléctrico de la ciudad de Alcalá de Henares. La Figura 1.7(b) muestra la escala de colores utilizada para la representación. De forma coherente con la Tabla 1.6, puede concluirse que el nivel de radiación de campo eléctrico en cualquier punto de Alcalá es muy bajo con respecto a los niveles de referencia establecidos por la ICNIRP.



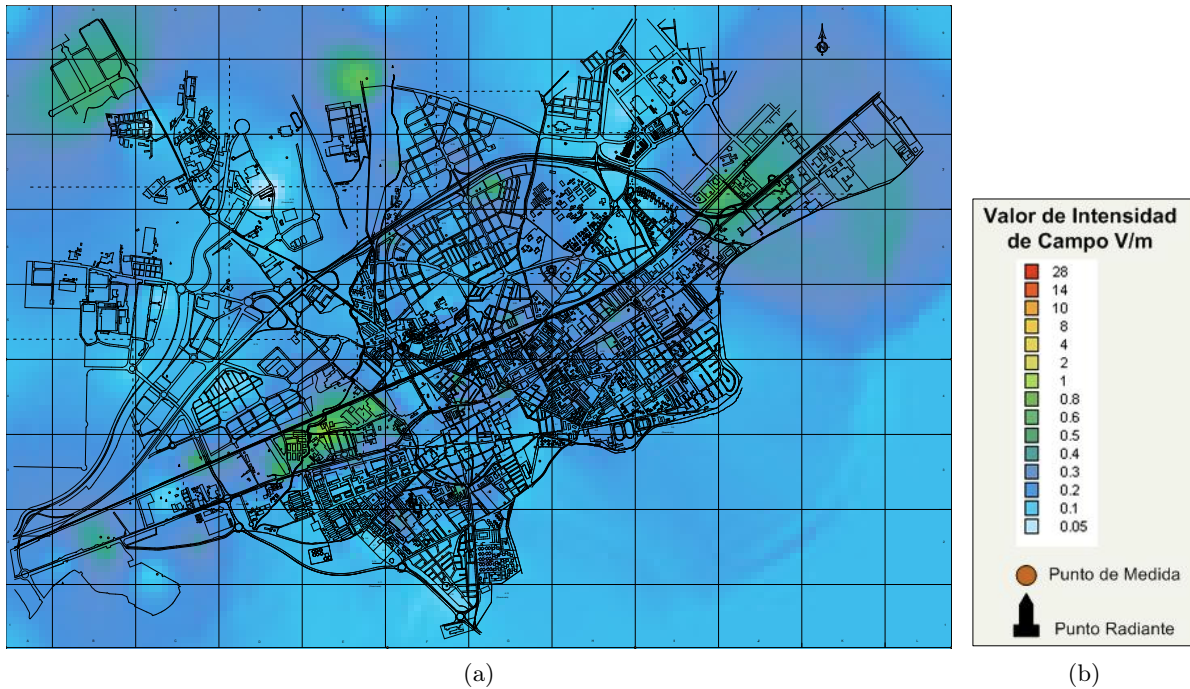
**Figura 1.6:** Mapa de la ciudad de Alcalá de Henares, sobre la que se ubican de forma manual 509 puntos de medida de emisión radioeléctrica.

**Tabla 1.6:** Valores máximos de las mediciones de campo eléctrico  $E(V/m)$  realizadas sobre los 8 distritos de la ciudad de Alcalá en el año 2006 para la frecuencia de 900 MHz en exposición de público general.

Distritos de Alcalá de Henares							
I	II	III	IV	V	VI	VII	VIII
0,34	0,37	0,84	0,54	0,50	0,92	1,24	0,82

## 1.4 Objetivos y estructura de la Tesis Doctoral

Este trabajo tiene dos objetivos fundamentales. El primero de ellos está centrado en la implementación de un algoritmo evolutivo que pueda resolver el problema de optimización *Mobile*



**Figura 1.7:** Mapa de emisiones radioeléctricas sobre la ciudad de Alcalá de Henares en el año 2006.

*Network Deployment Problem* (MNDP) con control de polución electromagnética. El problema MNDP consiste en identificar cierto número de localizaciones para la instalación de estaciones base componentes de una red de telecomunicación en una zona establecida. Se trata de un problema de optimización porque se pretende hallar una solución que optimice los recursos de la red (menor coste posible con un grado de cobertura mínimo aceptable). Este problema ha sido abordado previamente en otros trabajos sin tener en cuenta el parámetro de radiación electromagnética. Cada una de las estaciones base emite radiación y la intención es reducir en lo posible esta radiación emitida por el conjunto de estaciones en la zona establecida. No hemos encontrado estudios de despliegue de redes con control de radiación hasta el momento y esta es una de las aportaciones principales de este trabajo.

El segundo objetivo consiste en el desarrollo e implementación de un algoritmo de optimización novedoso y su aplicación al problema MNDP. Este algoritmo es el *Coral Reefs Optimization* y ha sido recientemente publicado en la literatura en [88, 89]. A lo largo de los capítulos experimentales de la tesis se evalúan el algoritmo evolutivo y el *Coral Reefs Optimization* aplicados en el problema MNDP. Además para medir la eficacia de los mismos, se implementan también otros algoritmos conocidos que resuelven el problema MNDP con control de polución electromagnética. Dichos algoritmos son: *greedy*, *Particle Swarm Optimization* (PSO) y *Harmony Search* (HS). El resto del documento está organizado de la siguiente manera:

- En el Capítulo de estado del arte describimos genéricamente las técnicas de *Soft Computing* y las aplicaciones de estas técnicas a problemas de posicionamiento de nodos en despliegue de redes de telecomunicación. A continuación estudiamos con mayor detalle tres algoritmos heurísticos en los que nos basaremos para resolver el problema MNDP. Estos son: el algoritmo evolutivo (EA), el algoritmo PSO y el algoritmo HS.
- El siguiente capítulo formula matemáticamente el problema MNDP que pretendemos re-

solver, planteando formalmente las variables del problema y la función de evaluación que tratamos de optimizar. Para esta última se desarrollan las expresiones matemáticas que permiten calcular el valor de las variables que la definen.

- El Capítulo 4 realiza una descripción detallada de los tres algoritmos descritos: EA, PSO y HS. Añadimos un cuarto algoritmo de tipo *greedy* con menor complejidad computacional que el resto. Dicho algoritmo nos permite delimitar una cota mínima a la calidad de las soluciones del resto de algoritmos. Estudiamos tanto las funciones comunes a todos ellos, como las funciones específicas de cada uno, aportando los pseudo-códigos correspondientes.
- En el Capítulo 5 se plantea un método de optimización novedoso, el algoritmo *Coral Reefs Optimization* o CRO, recientemente publicado en [88, 89]. Este algoritmo está inspirado en el comportamiento de los arrecifes de coral y en su evolución a través del tiempo. En la primera parte del capítulo exponemos la metodología llevada a cabo por el CRO, mientras que en la segunda parte describimos detalladamente las funciones del mismo.
- En el Capítulo 6 se redactan los resultados obtenidos con los cinco algoritmos presentados en los capítulos anteriores: EA, *Greedy*, PSO, HS y CRO. Primero exponemos las especificaciones para el problema MNDP y los valores tanto comunes como específicos para todos los algoritmos. Después pasamos al análisis de resultados de cada uno de ellos y un estudio comparativo de los cinco.
- En el último Capítulo recogemos las conclusiones del trabajo realizado, las aportaciones de la Tesis Doctoral como trabajo de investigación y las líneas de trabajo futuras pendientes por investigar. Cerramos esta sección con la lista de publicaciones a la que ha dado lugar este trabajo.
- Incluimos dos Anexos al final del documento. El Anexo A ofrece una breve exposición sobre los tipos de radiación dependiendo de la frecuencia empleada.
- En el Anexo B aportamos una primera versión del algoritmo evolutivo modificado. Esta modificación consiste en dos nuevas consideraciones. Por una parte se añade la variable de capacidad máxima de conexión para las estaciones base. Por otra, tres conjuntos de puntos o ubicaciones definidos en el planteamiento matemático se consideran ahora disjuntos. Estos conjuntos son los puntos de medición de campo eléctrico inicial, las posibles ubicaciones para la instalación de estaciones base y los puntos de interés donde disfrutar del servicio de telecomunicación. Hemos elegido un algoritmo genético de agrupamiento (*grouping genetic algorithm* o GGA) en el que cada localización está asignada a una única estación base, aun en el caso de que sean varias las que pudieran entregarle la capacidad demandada.



## Capítulo 2

# Estado del arte

En este capítulo realizamos un breve análisis de las diferentes técnicas de *soft-computing* desarrolladas para la resolución de problemas de optimización. Específicamente, en las correspondientes secciones se estudian con mayor detalle tres métodos clásicos de *soft-computing* aplicados en este trabajo para la resolución del problema MNDP (*Mobile Network Deployment Problem*). Estas técnicas son: el algoritmo evolutivo (*Evolutionary Algorithm* o EA), el algoritmo de *Particle Swarm Optimization* (PSO) y el algoritmo de *Harmony Search* (HS).

### 2.1 Técnicas de soft-computing

El término *Soft Computing* nació en los años 90, L. A. Zadeh lo definió en 1994 bajo los siguientes términos en [110]:

*“In traditional-hard-computing, the prime desiderata are precision, certainty, and rigor. By contrast, the point of departure in soft computing is the thesis that precision and certainty carry a cost and that computation, reasoning, and decision making should exploit, wherever possible, the tolerance for imprecision and uncertainty. ... In effect, in raising the banner of “Exploit the tolerance for imprecision and uncertainty”, soft computing uses the human mind as a role model and, at the same time, aims at a formalization of the cognitive processes humans employ so effectively in the performance of daily tasks. ... At this juncture, the principal constituents of soft computing are fuzzy logic (FL), neural network theory (NN), and probabilistic reasoning (PR), with the latter subsuming belief networks, genetic algorithms, parts of learning theory, and chaotic systems. ... FL is primarily concerned with imprecision, NN with learning, and PR with uncertainty.”*

Donde se contrastan los fines del *hard-computing*, que son la precisión y certeza o seguridad, con los del *soft-computing*, que desarrolla el cálculo, razonamiento y toma de decisiones, reduciendo el coste a cambio de otorgar cierta tolerancia a la imprecisión e incertidumbre. El *soft-computing* imita el modelo de pensamiento humano. El ser humano lleva a cabo gran diversidad de tareas inmerso en el mundo de incertidumbre e imprecisión en el que vive, actividades como la comprensión de un mensaje no carente de ruido, el reconocimiento o clasificación de imágenes o la conducción de vehículos. [110] describe los componentes del *soft-computing* como la lógica borrosa, las redes neuronales y el razonamiento aproximado, incluyendo en este último los algoritmos genéticos (bio-inspirados), redes bayesianas y teoría del caos. La siguiente relación de campos son algunos de los que abarca el término *Soft Computing*:

- Sistemas de *lógica difusa*,

- Redes neuronales,
- Computación bio-inspirada:
  - Algoritmos evolutivos,
  - Inteligencia de enjambre (Optimización de colonias de hormigas, alineamiento de aves en vuelo, ...)
- Redes bayesianas,
- Teoría del caos.

En este trabajo nos centraremos en los algoritmos bio-inspirados. Estos son una clase de heurísticos. El término “heurística” tiene su origen en la palabra griega “heuriskein”, que significa: “hallar, inventar”. La utilización de métodos heurísticos es recomendable cuando acontecen una o varias de estas circunstancias:

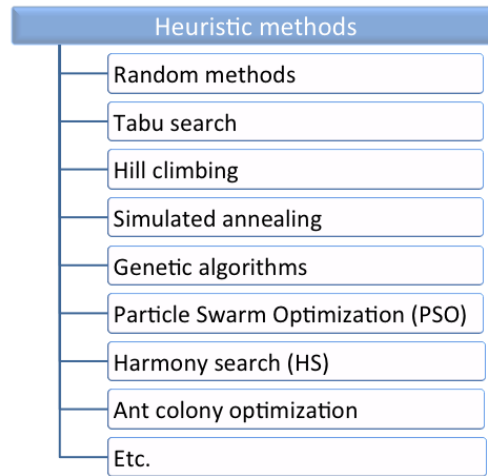
- Cuando no se puede aplicar otro método, en general por falta de hipótesis para aplicar algoritmos determinísticos.
- Cuando existe ruido o el problema está mal condicionado, los algoritmos heurísticos tienden a ser robustos y poco sensibles al ruido.
- Cuando existen numerosos óptimos locales, donde los algoritmos tradicionales basados en derivadas quedan atrapados.
- En problemas de gran complejidad, donde es suficiente encontrar una buena solución aunque no sea necesariamente el óptimo global.

Los métodos heurísticos pueden clasificarse atendiendo a diferentes criterios. Dependiendo del número de soluciones que gestionen en cada acción, los métodos heurísticos pueden ser secuenciales o paralelos. Los heurísticos secuenciales consideran y tratan una única solución en cada paso del algoritmo, mientras que las técnicas paralelas manejan diversas soluciones a la vez. Si atendemos al número de variables del problema que se desea optimizar, tenemos los algoritmos mono-objetivo, que persiguen optimizar un único parámetro, y los algoritmos multi-objetivo, donde se pretende optimizar varios parámetros que definen el problema y que casi siempre son dependientes unos de los otros, por lo que es necesario llegar a un compromiso entre ellos. Se dice que el algoritmo heurístico es evolutivo cuando su ejecución está basada en la evolución de etapas sucesivas para una o varias soluciones. En caso contrario se denomina heurístico no evolutivo. Por último, si el método descarta siempre cualquier solución que no mejore la actual, hablamos de un heurístico *greedy*, mientras que si soluciones de este tipo son aceptadas con cierta probabilidad, calificaremos al heurístico como no *greedy*.

La Figura 2.1 clasifica los métodos heurísticos en seis categorías que a continuación describimos brevemente:

1. Métodos aleatorios: son algoritmos secuenciales, de tipo *greedy* y no evolutivos. Estas técnicas van muestreando poco a poco el espacio de soluciones acotando las zonas no exploradas y registrando en cada momento la mejor solución encontrada.





**Figura 2.1:** Clasificación de métodos heurísticos. Este trabajo se centra especialmente en los algoritmos genéticos, PSO y HS.

2. Métodos *tabu search* o búsqueda tabú: éstos son de tipo paralelo, evolutivo y no *greedy*. La búsqueda tabú fue presentada en [49] y constituye el principal metaheurístico dentro de lo que se conoce como Programación mediante Memoria Adaptativa (AMP). Se caracterizan por ser métodos de búsqueda por entornos en los cuales se utiliza información acerca de los movimientos realizados con anterioridad. Las soluciones vecinas a las soluciones que se exploran quedan registradas en una lista tabú para evitar estudiar de nuevo zonas ya exploradas en el espacio de soluciones. De este modo se evita caer en ciclos durante la búsqueda por entornos. La lista tabú es dinámica, tras cierto número de iteraciones, los movimientos prohibidos dejan de serlo. A veces se permiten soluciones no mejores que las existentes para escapar de óptimos locales, por ello se les caracteriza como métodos no *greedy*.
3. Técnicas de *hill climbing*: son de tipo secuencial, no evolutivo y *greedy*. Una solución es sustituida por otra si y solo si ésta es mejor que la primera. Este proceso se realiza hasta no poder mejorar más la solución, es decir, no poder ascender más en el escalado. Los algoritmos *hill climbing* suelen detenerse en óptimos locales.
4. Algoritmos de tipo *simulated annealing* o recocimiento simulado: son secuenciales, evolutivos y no *greedy*. Este tipo de métodos están inspirados en la forma de obtención de ciertas aleaciones en forja: una vez fundido el metal, se va enfriando poco a poco hasta conseguir finalmente la estructura cristalina con la dureza y resistencia deseadas. Pueden estudiarse más ampliamente en [29, 66]. Para evitar los mínimos locales no siempre se acepta la solución óptima, sino que con cierta probabilidad puede escogerse otra solución menos óptima que la primera, siempre y cuando la diferencia entre ambas sea inferior a un umbral determinado.
5. Algoritmos genéticos: trabajan con una población de individuos que compiten por alcanzar la siguiente generación, por tanto, son de tipo paralelo y evolutivo [26]. Están basados en la teoría de evolución de las especies, en la que los individuos mejor adaptados al

entorno tienen mayores posibilidades de sobrevivir y tener descendencia transmitiendo sus características genéticas a futuras generaciones asegurando así la continuación de su especie con los especímenes mejor adaptados. En este símil, cada individuo representa una solución al problema y la carga genética del individuo es precisamente el valor de los parámetros que definen la codificación de la solución. El hecho de hablar de probabilidad de supervivencia dependiendo del grado de adaptación hace que estos algoritmos sean de tipo no *greedy* puesto que factores distintos de la carga genética pueden favorecer que un individuo pase a la generación siguiente, como por ejemplo circunstancias fortuitas. La adaptación de un individuo se mide como el valor de una función de evaluación o función objetivo. Dicho valor se conoce como valor de *fitness*. Un conjunto de operadores intervienen en todas y cada una de las generaciones, éstos son los operadores de selección de individuos, reproducción y mutación. La condición de fin o parada del algoritmo suele consistir en alcanzar una solución lo suficientemente buena, o bien ejecutar tras un número máximo de generaciones, o bien que el algoritmo no progrese durante cierto número consecutivo de generaciones. En apartados posteriores se realiza una exposición detallada del algoritmo genético ya que es uno de los algoritmos implementados para la resolución del problema MNDP (*Mobile Network Deployment Problem*).

6. Optimización por cúmulos de partículas (*Particle Swarm Optimization* o PSO): son algoritmos de tipo paralelo, evolutivo y no *greedy*. Se trata de una técnica meta-heurística basada en poblaciones e inspirada en el comportamiento social del vuelo de las bandadas de aves o el movimiento de los bancos de peces. El PSO fue originalmente desarrollado conjuntamente por el psicólogo-sociólogo James Kennedy y por el ingeniero electrónico Russell Eberhart [63], basándose en un enfoque conocido como la metáfora social. La metáfora social se puede resumir de la siguiente forma: los individuos que conviven en una sociedad tienen una opinión que es parte de un conjunto de creencias (el espacio de búsqueda) compartido por todos los posibles individuos. Cada individuo puede modificar su propia opinión basándose en tres factores: su conocimiento sobre el entorno (valor de *fitness*), su conocimiento histórico o experiencia personal anterior (su memoria) o bien el conocimiento histórico o experiencias anteriores de los individuos en su vecindario (relación social). El funcionamiento básico del PSO simula el comportamiento del vuelo de las bandadas de aves o bancos de peces en busca de comida. La estrategia más eficaz es seguir al ave que está más cerca de la comida. Cada ave se modela como una partícula solución en el espacio de búsqueda que está siempre en continuo movimiento y nunca se elimina o muere. El PSO es un sistema multiagente, es decir, las partículas son agentes simples que se mueven por el espacio de búsqueda y que guardan la mejor solución que han encontrado. Este algoritmo también ha sido implementado para la resolución del problema MNDP que nos ocupa y en capítulos posteriores hablaremos de él nuevamente.
7. La búsqueda en armónica o *Harmony Search* (HS) es una de las técnicas heurísticas más recientes. Es de tipo paralelo, evolutivo y no *greedy*. Fue presentada en el año 2001 por [45] y trata de imitar el proceso de improvisación de una orquesta de jazz que trata de componer una melodía que guste al público. La analogía es la siguiente [73]: cada nota musical representa una variable de decisión y una partitura completa se corresponde con una posible solución del problema. El valor de la función objetivo a minimizar viene dado por la combinación de las notas musicales que forman la partitura. En cada iteración, los músicos tratan de mejorar la calidad de la misma. Si la partitura creada es mejor que la peor partitura almacenada en una memoria, entonces la primera ocupa su lugar en la memoria. Existen diferentes variables de ajuste del algoritmo como son el ancho de banda

o el *Pitch Adjustment Rate* (PAR), que tiene cierta semejanza con el operador de mutación de un algoritmo genético. También este algoritmo se ha desarrollado para la aplicación del problema MNDP en este trabajo.

8. Los métodos de optimización por colonia de hormigas son de tipo paralelo, evolutivo y no *greedy* [31]. Estas técnicas fueron presentadas por primera vez en [30]. Están basados en el comportamiento de las hormigas, donde estos animales casi ciegos y moviéndose más o menos al azar, pueden encontrar el camino más corto desde su nido hasta la fuente de alimentos y regresar. Una hormiga al desplazarse dispersa una sustancia química (denominada feromona) que atrae a otras hormigas para que puedan seguir su rastro. En principio, una hormiga aislada se mueve de forma aleatoria, eligiendo en cada bifurcación un camino al azar. Las siguientes hormigas construyen su camino a partir de la cantidad de feromonas que detectan en cada bifurcación. Ante poca cantidad de feromonas seguirán una ruta aleatoria.

Durante los últimos años se han desarrollado de forma exitosa un elevado número de procedimientos heurísticos. Estos se intentan aplicar progresivamente a problemas cada vez de mayor envergadura y alcance. Así ha surgido el término “metaheurística”, donde el prefijo “meta-” hace referencia a un nivel superior. El término “metaheurístico” se define formalmente en [78] como:

*“Un metaheurístico es un procedimiento iterativo, con una estructura y unas reglas generales de funcionamiento que lo caracterizan, que guía un método subordinado (normalmente un heurístico) combinando de forma inteligente diversos conceptos para explorar los espacios de búsqueda utilizando estrategias aprendidas para conseguir soluciones cuasi-óptimas de manera eficiente.”*

De la misma manera en [50] se define la metaheurística de la siguiente forma:

*“Metaheurística se refiere a una estrategia maestra que guía y modifica otras heurísticas para producir soluciones más allá de aquellas que normalmente se generan en una búsqueda de óptimos locales.”*

Tradicionalmente se clasificaban los algoritmos metaheurísticos en dos grandes grupos: metaheurísticos basados en población y metaheurísticos basados en búsqueda por entornos. Sin embargo, en la actualidad se trabaja con algoritmos híbridos de forma que los algoritmos basados en población incluyen técnicas de búsqueda local basadas en entornos.

También encontramos una clasificación de metaheurísticos dependiendo de la toma de decisión: algoritmos aleatorios, determinísticos y basados en memoria (soluciones halladas en etapas anteriores durante su ejecución). En la práctica los algoritmos actuales tienen rasgos de cada grupo, con estrategias pseudo-aleatorias.

En los últimos tiempos se ha utilizado el término de algoritmos *hiperheurísticos* haciendo referencia a aquellos heurísticos que eligen entre diversas técnicas heurísticas para resolver diferentes problemas de optimización. Aquí, el usuario puede combinar varias heurísticas buscando combinar las ventajas que aportan y/o compensar los inconvenientes de alguna frente a otras en el problema a resolver [102].

## 2.2 Introducción a los algoritmos genéticos

En esta sección realizamos una breve exposición sobre la teoría de los algoritmos genéticos. Los algoritmos genéticos pertenecen al grupo de los métodos heurísticos evolutivos. Desde su origen

en 1975 por el investigador John Holland [55] han sido ampliamente desarrollados. Son algoritmos bio-inspirados basados en la teoría de evolución de las especies [35], donde los individuos mejor adaptados al entorno tienen mayores posibilidades de sobrevivir y tener descendencia, transmitiendo así sus características genéticas a futuras generaciones. De esta forma no solo se asegura la continuidad de la especie sino que además ésta va adaptándose a los cambios del entorno con los mejores especímenes de cada generación. En este símil, cada individuo representa una solución del problema de optimización a resolver y la carga genética del individuo es precisamente el valor de los parámetros que definen la solución.

El algoritmo trabaja con una población de individuos que compiten por alcanzar la siguiente generación, es por tanto, un algoritmo evolutivo y paralelo. El hecho de hablar de probabilidad de supervivencia del individuo dependiendo del grado de adaptación hace que el algoritmo sea de tipo no *greedy*. Factores distintos a la propia carga genética, como por ejemplo circunstancias fortuitas, pueden favorecer que un individuo prospere hasta una generación posterior. La adaptación de un individuo se mide mediante el valor alcanzado en la denominada función de evaluación, también conocida como función *fitness* o función objetivo. Dependiendo del problema, la optimización consiste bien en maximizar o bien en minimizar dicha función. La función de evaluación permite discriminar entre dos individuos, determinando cuál de ellos es el mejor adaptado, es decir aquél con el valor de *fitness* más cercano al máximo/mínimo de la función.

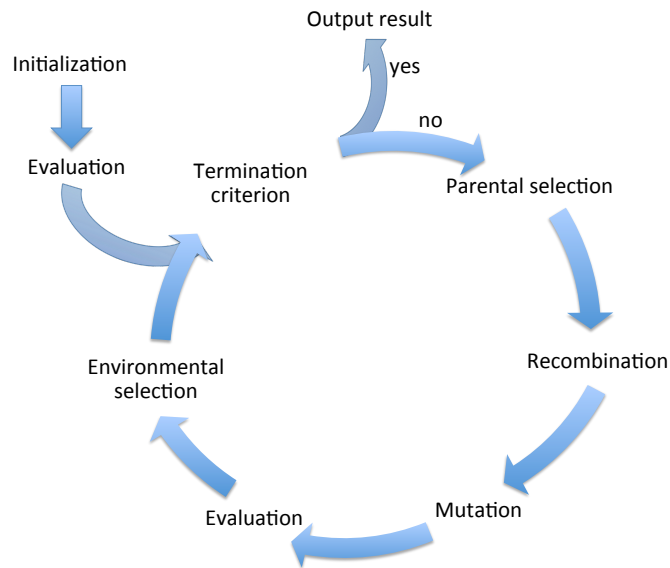
La implementación de un algoritmo genético está caracterizada por tres operadores básicos [3]: el operador de selección de individuos para componer la población de una generación y/o determinar los individuos que tendrán descendencia, el operador de cruce o recombinación para la obtención de individuos descendientes de individuos progenitores y el operador de mutación, que con reducida probabilidad, provoca cambios sustanciales en la composición de los individuos. Las consecuencias de la mutación en un individuo pueden resultar desastrosas para su supervivencia, o bien al contrario, favorecerla de manera muy ventajosa.

La Figura 2.2 muestra el diagrama de flujo básico de un algoritmo genético. El algoritmo comienza con un conjunto de individuos aleatorios que componen la población inicial. La población inicial representa un conjunto de soluciones del problema. Siendo éstas aleatorias, lo esperable es que tras evaluarlas no resulte ninguna de buena calidad, es decir, con valores de *fitness* alejados del óptimo. Si no se cumplen las condiciones de terminación del algoritmo se progresa a la siguiente generación. En cada generación se procesa la población con los operadores de selección de padres, cruce y mutación. Después se evalúan los nuevos individuos y se realiza la selección de supervivientes para la generación inmediata posterior. En caso de no satisfacer de nuevo las condiciones de parada se da paso a una nueva generación.

Este diagrama básico acepta otros procedimientos como la reparación de individuos y/o la búsqueda local. La reparación de individuos es necesaria cuando la codificación de las soluciones permite que un individuo no se corresponda con una solución viable. Entonces es preciso modificar o reparar el individuo para que éste represente a una única solución factible. El proceso de búsqueda local consiste en realizar pequeños cambios en el individuo tal que resulte otro muy similar al primero pero de mayor calidad. La función de búsqueda local aumenta notablemente la carga computacional, de modo que se suele introducir por ciclos de generaciones. Incluir en el algoritmo genético este tipo de técnicas convierten al algoritmo en un método híbrido. Por ejemplo, un algoritmo memético puede entenderse como un algoritmo genético al que se le ha agregado la acción de búsqueda local después del operador mutación en la Figura 2.2.

Los criterios de parada o fin de ejecución del algoritmo son definidos generalmente como uno o varios entre los siguientes:

- Evolución de un número máximo de generaciones.



**Figura 2.2:** Diagrama de flujo básico de un algoritmo genético. En cada generación se realizan operaciones de evaluación o cómputo de la función de *fitness*, selección, cruce y mutación.

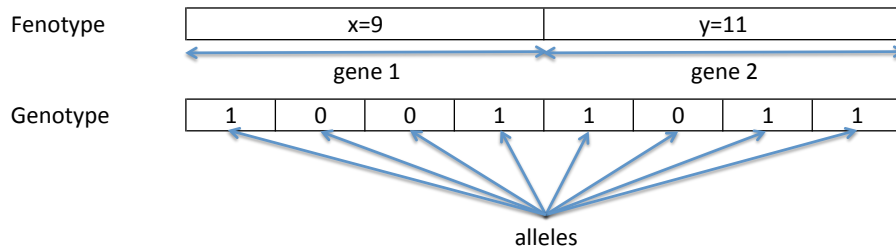
- Alcanzar cierto número de generaciones en los que no se detecta una mejora apreciable en la población. Tras sucesivas generaciones el valor medio del *fitness* en la población se permanece constante, entonces se dice que el algoritmo no ha progresado en dichas generaciones.
- Encontrar una solución suficientemente próxima a una óptima previamente acotada.

A continuación tratamos en detalle los tres operadores básicos mencionados junto con asuntos de gran importancia para la eficacia del algoritmo como la codificación de la solución y el proceso de búsqueda local. Una explicación más pormenorizada sobre la teoría de algoritmos genéticos se puede encontrarse en [27, 38, 51, 74].

### 2.2.1 Codificación de la solución

El proceso de codificación permite la representación de las diferentes soluciones del problema real en individuos que participan en el algoritmo. Cada individuo se representa en forma de un vector numérico (generalmente se utiliza números binarios, pero también pueden ser números enteros o reales dependiendo de las características del problema). El vector de números que define un individuo se denomina cromosoma, mientras que cada elemento que lo compone se le asigna el nombre de alelo. Se llama gen a un conjunto de alelos que codifican una misma característica o parámetro de la solución. La Figura 2.3 muestra un ejemplo de codificación binaria, donde el cromosoma (10011011) representa a la solución  $\{x = 9; y = 11\}$ . El cromosoma está compuesto por dos genes (el gen (1001) hace referencia a la variable  $x$  y la segunda mitad del cromosoma o gen (1011) se corresponde con la variable  $y$ ). Cada bit que compone un gen es un alelo. El cromosoma (10011011) tiene por tanto ocho alelos.

Se define genotipo (*Genotype* o *Genotypic space*) al espacio de individuos, mientras que el fenotipo (*Phenotype* o *Phenotypic space*) es el espacio de soluciones al problema real. En el ejemplo de la Figura 2.3, el vector (10011011) corresponde al genotipo, mientras que la solución  $\{x = 9; y = 11\}$  hace referencia al fenotipo.



**Figura 2.3:** Ejemplo de codificación binaria. La codificación de la solución se conoce como genotipo. El individuo o cromosoma se compone de un número determinado de genes. El fenotipo se corresponde con la solución que representa el cromosoma.

Habitualmente, los genes de un cromosoma son independientes entre sí en el sentido en que el valor que puede tomar un gen no depende de los valores que adquieren otros genes. Sin embargo, existen casos en los que sí se evidencian este tipo restricciones, como por ejemplo en problemas en los que la solución es precisamente el cómo ordenar un conjunto de elementos (cada permutación de genes en el cromosoma es una solución diferente) como es el caso del clásico problema del viajante (*Travelling Salesman Problem*). Para asegurar el cumplimiento de estas dependencias puede ser necesario reparar aquellos individuos que no cumplan las condiciones de dependencia y transformarlos en otros muy similares que además respeten las restricciones impuestas.

Diferenciamos dos tipos de codificación: directa e indirecta. Un algoritmo utiliza una codificación directa cuando cada individuo representa una solución en la realidad de forma inmediata y sin necesidad de realizar transformación alguna. El individuo contiene en sí mismo toda la información necesaria para implementar la solución. Si se requiere de un proceso de conversión para pasar del vector solución (genotipo) a la solución en la realidad (fenotipo), se dice que la codificación es de tipo indirecto. Generalmente, la codificación indirecta es computacionalmente más costosa por el proceso de traducción, conversión o acceso a tablas de datos externos. Si bien es más efectiva porque permite tener mayor conocimiento del problema. Un ejemplo de codificación indirecta es el problema del viajante manejando datos externos como la información sobre las ciudades en cuanto a su posición, ciudades vecinas, distancias entre las mismas, etc.

Existe otra clasificación para la codificación de un algoritmo, dependiendo si el orden en el que aparecen los valores para los genes define unívocamente a una solución o no:

- Codificaciones basadas en el orden, donde el orden en el que aparecen las variables es muy importante para identificar cada solución. Entre éstas se encuentra la codificación por permutaciones, como es el caso del problema del viajante, y otras codificaciones empleadas por ejemplo, para la resolución de sudokus.
- Codificaciones no basadas en el orden: el orden de las variables no es importante para la solución en sí misma, que queda identificada por el valor de cada gen. Existen cuatro tipos de codificaciones no basadas en el orden:
  - Codificación binaria. Puede tratarse bien de la codificación binaria común (las variables son binarias y por lo tanto adoptan uno de dos valores posibles) o bien de codificación binaria de Gray en el que dos valores sucesivos para una variable difieren únicamente en un dígito. Por ejemplo, un código Gray de dos bits define un conjunto de cuatro valores: {00, 01, 11, 10}

- Codificación con valores enteros, el contenido de los genes pertenece al conjunto de números enteros.
- Codificación con valores reales, análogo al anterior pero las variables solo pueden tomar valores en  $\mathbb{R}$ .
- Codificación con valores finitos, donde las variables pueden tomar solo valores pertenecientes a un conjunto limitado de valores, como puede ser un conjunto de colores predefinidos, un conjunto cerrado de posiciones en un tablero o un conjunto  $\mathcal{T}$  de elementos que definen el tipo de equipos a utilizar.

La elección de una u otra codificación se encuentra muy ligada al tipo de problema sobre el que se desea aplicar. Antes de determinar la forma de representación de las soluciones es conveniente tener en cuenta las características que comentamos a continuación:

- Todas las posibles soluciones al problema deben tener representación en el conjunto de individuos. Resultaría inaceptable de otra manera, la totalidad de las soluciones posibles deben poderse codificar.
- Todos los individuos posibles deben corresponderse con soluciones factibles. A veces esto no sucede y aparecen individuos que no representan soluciones viables o válidas. En estas ocasiones se realiza un proceso de reparación del individuo para transformarlo en otro individuo similar pero que sí represente a una solución factible del problema. Es importante considerar que la operación de reparación, en caso de ser necesaria, se realizaría un gran número de ocasiones en cada generación. Por lo tanto, en cuanto a tiempo de ejecución del algoritmo, resulta mucho más ventajoso elegir una codificación que no requiera reparaciones de individuos frente a otra codificación que sí las precise.
- En general, la parte computacionalmente más costosa de un algoritmo genético es el cálculo de la función de evaluación. Este cálculo se realizará como mínimo  $n$  veces por generación, donde  $n$  es el número de nuevos individuos. Para calcular el valor de *fitness* de un individuo es necesario realizar la transformación del mismo a la solución que éste represente. La elección de uno u otro tipo de codificación puede resultar clave para que el cálculo de la función de evaluación sea sencillo, rápido y directo, o por el contrario, resulte tedioso y costoso en términos de computación.
- De la misma manera, el operador de selección de individuos es otra de las partes más costosas en tiempo de computación. La razón fundamental es que este proceso está basado en la comparación de los valores de *fitness*.
- La codificación debiera facilitar en lo posible las operaciones de cruce y mutación. Si bien suelen ser operadores más simples que la selección, también se ejecutan un número muy elevado de veces en cada generación. De modo que, cuanto más sencillas y rápidas sean éstas, más veloz resultará el algoritmo.
- Por último y como se menciona en [82], el método de codificación elegido para un algoritmo genético influye en gran medida en la capacidad del dicho algoritmo para encontrar soluciones de alta calidad. Especialmente afecta sobre el parámetro denominado localidad. El nivel de localidad está muy relacionado con el modo en que una solución (fenotipo) es alterada al aplicar pequeños cambios sobre su cromosoma (genotipo). Mayor nivel de localidad implica menor grado de modificación de la solución tras los cambios realizados sobre el individuo. En general se considera que mayor nivel de localidad favorece la búsqueda de

soluciones de más alta calidad porque de forma intuitiva, significa que el algoritmo busca en los alrededores muy próximos a una solución sobre el espacio de soluciones.

### 2.2.2 Función de evaluación

La función de evaluación o función objetivo es aquella que se utiliza para cuantificar el grado de aptitud de los individuos. De la correcta definición de esta función dependerá la eficacia del algoritmo para encontrar soluciones de alta calidad.

En ocasiones la función de evaluación considera diferentes variables simultáneamente. Generalmente las variables de la función de evaluación se corresponden con magnitudes diferentes e incluso tienen intereses contrapuestos. Un ejemplo sencillo de una red de telecomunicación móvil, donde la solución óptima es la más económica posible siempre que no supere un porcentaje máximo de zonas sin servicio o cobertura de red. El coste de la solución será tanto más alto cuanto menor sea el porcentaje total de zonas sin cobertura. La solución en estos casos es encontrar un punto de compromiso entre los valores asignados a ambas variables, encontrar la solución de menor precio posible con un porcentaje de servicio mínimo requerido.

En la mayoría de los casos, las variables que intervienen en la función de evaluación son magnitudes independientes entre sí. En el ejemplo anterior se combina el coste total de la red de comunicación,  $\mathcal{C}_{\mathcal{T}}$ , medido en euros (€) con el porcentaje total de zonas sin cobertura,  $P_{no\ cober}$  (adimensional). Para combinar ambas variables en una misma expresión matemática, se incorporan coeficientes que permitan la comparación de los valores adimensionales. La Ecuación 2.1 muestra la función objetivo para el ejemplo mencionado. El algoritmo evolutivo trataría de encontrar un mínimo para esta función, donde  $k_1$  (€<sup>-1</sup>) es el coeficiente de la variable coste  $\mathcal{C}_{\mathcal{T}}$  y  $k_2$  es el coeficiente del porcentaje total de zonas sin cobertura  $P_{no\ cober}$ . Asignado unos u otros valores a dichos coeficientes conseguiremos que las soluciones encontradas por el algoritmo optimicen más una magnitud u ambas, es decir si  $k_1 \cdot \mathcal{C}_{\mathcal{T}}$  es mayor que  $k_2 \cdot P_{no\ cober}$  se centrará en la búsqueda de soluciones muy económicas, en caso contrario buscará soluciones con gran porcentaje de cobertura a pesar del coste que esto suponga. Lo ideal es ajustar el valor de los coeficientes para alcanzar el grado de compromiso deseado entre los distintos objetivos.

$$\mathcal{F} = k_1 \cdot \mathcal{C}_{\mathcal{T}} + k_2 \cdot P_{no\ cober} \quad (2.1)$$

### 2.2.3 Selección de individuos

Según comentamos en la Figura 2.2 podemos decir que existen dos momentos claramente diferenciados donde interviene el proceso de selección de individuos en un algoritmo genético: el instante en el que se forma la población para la generación siguiente y en el paso previo a la recombinación o cruce de individuos. Es en la arquitectura empleada donde se define la selección de individuos, la forma y los porcentajes sobre la población total. Por ejemplo, en algunos casos, no todos los individuos participan en la recombinación como progenitores. En otros, el proceso de selección puede permitir que un individuo sea elegido varias veces como progenitor en diferentes parejas. A veces se desglosa la población en dos conjuntos y cada uno de ellos está destinado a alimentar a uno de los dos operadores (cruce o mutación). Otras veces la población completa participa en la recombinación. Generalmente parte de descendencia generada tras el cruce se procesa también en la función de mutación. En la mayoría de los casos se aplica elitismo, lo cual significa que al mejor individuo se le permite pasar directamente a la siguiente generación evitando que sea mutado fortuitamente. De esta forma aseguramos la supervivencia del mejor individuo de una población hasta la siguiente generación.



La selección de individuos está basada en la función de evaluación de forma que se otorga mayor probabilidad de selección a los individuos con mejor valor de *fitness*, es decir, a los individuos mejor adaptados. Sin embargo la aleatoriedad no impide que algunos individuos de peor calidad superen el proceso de selección. La eficiencia de un operador de selección se mide de acuerdo a su grado de presión selectiva y su grado de diversidad, siendo ambos factores contrapuestos:

- Presión selectiva (*selective pressure*): permite que los mejores individuos sean seleccionados para el proceso de recombinación. Es necesaria para que el proceso de búsqueda no sea aleatorio y exista cierto grado de convergencia, centrando la búsqueda en regiones prometedoras.
- Diversidad: pretende evitar la convergencia prematura (convergencia rápida hacia zonas que no contienen el óptimo global sino algún óptimo local). La diversidad está asociada a las diferencias entre los individuos de una población. La falta de diversidad genética implica que todos los individuos en la población sean similares, sus descendientes también lo sean y el algoritmo progrese muy despacio o simplemente no progrese. La falta de diversidad hace que casi toda la población esté concentrada en un área del espacio de soluciones y el algoritmo no pueda salir de dicho área buscando nuevos espacios sin explorar.

Las técnicas de selección pueden clasificarse en tres grandes grupos [21]:

- Selección proporcional: los individuos se eligen de acuerdo a la contribución de su valor de *fitness* con respecto a las contribuciones del resto de la población. Esta técnica, presentada originariamente por [55], fue desarrollada posteriormente por [52]. Aparecieron entonces cuatro variantes: la ruleta, el sobrante estocástico, la universal estocástica y el muestreo determinístico; esta última con técnicas modificadas como el escalamiento *sigma*, el uso de jerarquías y la selección de Boltzmann. Vamos a prestar atención a la ruleta y al sobrante estocástico ya que son los más ampliamente utilizados. Ambos están se exponen en las Secciones 2.2.3.1 y 2.2.3.2.

- Selección mediante torneo: esta técnica fue propuesta por [105] y posteriormente estudiada con detalle en la tesis de [10]. El fundamento de esta técnica consiste en seleccionar en base a comparaciones directas de individuos. Se seleccionan aleatoriamente un número  $p$  de individuos. Típicamente se utiliza torneo por parejas de individuos siendo  $p = 2$ . De cada conjunto gana el individuo con mejor valor de *fitness*, es decir, el de mejor aptitud. Este proceso se repite  $N$  veces hasta completar la selección de  $N$  individuos. Una variante de este método consiste en asignar cierta probabilidad de éxito al individuo más apto de cada conjunto. De esta forma el grado de aptitud no garantiza siempre el ser seleccionado.

A medida que aumenta el tamaño del conjunto  $p$  el proceso de selección se va haciendo más elitista. El caso más extremo se alcanza cuando el conjunto  $p$  coincide con la población total. En estas circunstancias, el ganador siempre es el individuo con mejor valor de *fitness*. Como ya mencionamos, el *elitismo* permite asegurar la supervivencia de una generación a la siguiente del individuo mejor adaptado. Consiste en transmitir siempre y en cada generación el mejor individuo íntegro asignándole un lugar preferente en vez de abandonarle a su suerte durante la selección de individuos. Para evitar su desaparición en la población, se le protege de cualquier mutación directa sobre él.

- Selección de estado uniforme: se trata de una técnica que no suele utilizarse en los problemas de optimización, de modo que no la desarrollaremos aquí. Fue propuesta por [106] para

algoritmos genéticos no generacionales, en los que en cada generación, solo unos pocos individuos son reemplazados por otros más aptos. Consiste en seleccionar los individuos más aptos y someterlos a los operadores de cruce y mutación. Los descendientes resultantes más aptos sustituirán a los peores individuos de la población general para la siguiente generación.

### 2.2.3.1 Método de selección por ruleta

El método de selección mediante ruleta consiste en asignar a cada individuo una probabilidad de selección proporcional a la posición que ocupe su valor de *fitness* en el intervalo útil de valores de *fitness* de la población existente. De esta forma, los individuos con mejor valor de *fitness* tienen mayores posibilidades de ser elegidos que los individuos con peores valores de la función de evaluación. Esta técnica alcanzó gran popularidad desde que fuera publicada en [51]. Es una estrategia sencilla pero poco eficiente puesto que su complejidad crece de modo  $O(n^2)$  siendo  $n$  el tamaño de la población. A continuación mostramos un ejemplo numérico de selección mediante la técnica de la ruleta. Supongamos aquí que el algoritmo trata de maximizar el valor de la función objetivo. Sea una población compuesta por seis individuos ( $N_{ind} = 6$ ) con identificadores del 1 al 6 y ordenados en la Tabla 2.1 de forma decreciente según su valor de *fitness*  $\mathcal{F}_i$ . La primera columna de la tabla indica el identificador de cada individuo  $i$ , mientras que la segunda columna recoge su valor de *fitness*  $\mathcal{F}_i$ . La suma de todos los valores de *fitness* en la población es 20 y se muestra en la última fila de la tabla. Establecemos el rango de *fitness* [1-20] siendo 1 el valor mínimo de *fitness* en la población y 20 la suma de todos los valores de *fitness*. Este intervalo será proporcional al rango de probabilidad de selección [0-1] para los individuos. La tercera columna de la tabla indica la probabilidad de selección  $p_i$  del individuo  $i$  mediante la Ecuación 2.2. El nombre del método de la “ruleta” hace referencia al reparto de probabilidades como si de una tarta o pastel se tratara. La Figura 2.4 representa la distribución de probabilidades por sectores circulares según la última columna de la Tabla 2.1, que expresa la probabilidad en el rango [0-360] correspondiendo gestionando así sectores circulares de la “tarta a repartir”.

**Tabla 2.1:** Obtención de probabilidades de selección de individuos mediante la técnica de la ruleta para población de seis individuos, teniendo en cuenta que el algoritmo genético tiene como objetivo maximizar la función de evaluación.

Indiv. $i$	$\mathcal{F}_i$	Prob. $p_i$	Ángulo del sector circular (°)
1	10	0,5	180
2	4	0,2	72
3	2	0,1	36
4	2	0,1	36
5	1	0,05	18
6	1	0,05	18
$\sum_{i=1}^6$	20	1	360

$$p_i = \begin{cases} \frac{\mathcal{F}_i}{\sum_{i=1}^{\mathcal{N}_{ind}} \mathcal{F}_i}; & i = 1 \\ p_{i-1} + \frac{\mathcal{F}_i}{\sum_{i=1}^{\mathcal{N}_{ind}} \mathcal{F}_i}; & 1 < i \leq \mathcal{N}_{ind} \end{cases} \quad (2.2)$$



**Figura 2.4:** Distribución de probabilidades de selección por el método de la ruleta según el ejemplo de la Tabla 2.1.

Cuando la optimización consiste en minimizar la función de evaluación es necesario transformar la Ecuación 2.2 en la Ecuación 2.3. En vez de trabajar con los valores de *fitness* de cada individuo  $\mathcal{F}_i$  se utilizan los inversos, es decir,  $\frac{1}{\mathcal{F}_i}$ . Tenemos que destacar que en estos casos no podemos tener individuos con valor nulo de *fitness* puesto que significaría dividir por cero en la Ecuación 2.3. En la Sección 4.1.4 se ofrece un ejemplo análogo al expuesto cuando se minimiza la función de *fitness*.

$$p_i = \begin{cases} \frac{1/\mathcal{F}_i}{\sum_{i=1}^{\mathcal{N}_{ind}} \frac{1}{\mathcal{F}_i}}; & i = 1 \\ p_{i-1} + \frac{1/\mathcal{F}_i}{\sum_{i=1}^{\mathcal{N}_{ind}} \frac{1}{\mathcal{F}_i}}; & 1 < i \leq \mathcal{N}_{ind} \end{cases} \quad (2.3)$$

### 2.2.3.2 Sobrante estocástico

Esta técnica fue propuesta por [7, 10]. Existen dos versiones: sobrante estocástico con resto, de complejidad  $O(n^2)$ , y sobrante estocástico sin resto y con complejidad  $O(n)$ . La selección con el método de sobrante estocástico mejora al método de la ruleta introduciendo una mayor presión selectiva. Pero a consecuencia de esta presión puede desencadenarse la convergencia prematura del algoritmo. Para cada individuo se calcula el valor esperado  $e_i$  como el cociente entre su valor de *fitness*  $\mathcal{F}_i$  y el *fitness* promedio de la población  $\bar{\mathcal{F}}$  (ver Ecuación 2.4). Del valor esperado de cada individuo se trata por separado su parte entera y su parte decimal. Los individuos cuya parte entera de  $e_i$  sea no nula son seleccionados para la nueva población con tantas copias como indique dicha parte entera.

$$e_i = \frac{\mathcal{F}_i}{\bar{\mathcal{F}}} \quad (2.4)$$

En el sobrante estocástico sin resto, las partes decimales se utilizan para sesgar el tiro de una moneda que determina si un individuo se selecciona de nuevo o no. En el sobrante estocástico con resto, las partes decimales o restos se emplean para dimensionar los segmentos de una ruleta de forma similar a lo explicado en la técnica de la ruleta. De esta forma los huecos libres en la nueva población se completan mediante una selección aleatoria de individuos de la población original con una probabilidad de selección proporcional a la representación que tiene la parte decimal de su valor esperado en toda la población.

### 2.2.4 Operador de cruce

El objetivo de la operación de cruce o recombinación es explorar el espacio de soluciones de forma inteligente. A partir de dos cromosomas o individuos se genera uno nuevo. En ocasiones el operador se implementa para obtener dos descendientes a la vez de cada pareja de individuos. En la composición del individuo descendiente, se trata de extraer, preservar y combinar parte del material genético de sus progenitores, y al mismo tiempo añadir secciones nuevas en el cromosoma que aporten diversidad a la población. De esta forma el algoritmo busca trabajar generación tras generación con una población cada vez de mejor calidad.

En algunos casos se desea que no toda la población se recombine, sino que solo un grupo seleccionado pueda hacerlo. El resto de individuos se les permite generalmente acceder al operador de mutación o simplemente competir con el resto por sobrevivir a la siguiente generación. El conjunto definitivo de progenitores que se recombinen suele ser agrupado por parejas aleatorias. Como hemos dicho, de cada pareja surgen dos descendientes, de los cuales suele descartarse uno, bien al azar o bien el de peor valor de *fitness*. Existen numerosos tipos de cruce, algunos de los cuales vamos a describir brevemente:

1. Cruce por un punto único (*single point crossover*): este método fue propuesto por [55], es el más sencillo y popular. Posteriormente surgieron diversas variantes. Consiste en seccionar ambos cromosomas progenitores por un punto aleatorio, quedando subdivididos cada uno de los cromosomas en dos partes. El individuo descendiente se construye a partir de una parte de cada cromosoma. La Figura 2.5 muestra un ejemplo de cruce por un punto. Hemos utilizado letras para el contenido del primer progenitor y números para el segundo, con la intención de destacar la procedencia de los fragmentos en el cromosoma descendiente. En el ejemplo, ambos progenitores tienen una longitud de siete alelos, y el punto de corte se localiza entre el cuarto y el quinto alelo. El descendiente se construye con el fragmento inicial progenitor número 1 y el fragmento final del cromosoma progenitor 2. Un segundo descendiente, no incluido en este ejemplo, se formaría con los fragmentos no utilizados, es decir, el inicial del segundo cromosoma y el fragmento final del primero.

Parent 1	A	B	C	D	E	F	G
Parent 2	1	2	3	4	5	6	7
Descendent	A	B	C	D	5	6	7

**Figura 2.5:** Ejemplo de la operación de cruce o recombinación con corte de punto único en un algoritmo genético. Cada cromosoma se fragmenta en dos partes y el individuo descendiente recibe un fragmento de cada progenitor.

Dependiendo de la codificación empleada, a veces no todos los puntos de un cromosoma son aptos como puntos de corte. Es el caso de cortar en algún punto intermedio de un gen, generando descendientes posiblemente con valor de algún gen no válido. También pueden obtenerse individuos descendientes no viables cuando los cromosomas son por definición de longitud variable y algunos puntos de corte en ellos generarían individuos no válidos. Es imprescindible controlar este tipo de situaciones y resolver los conflictos que surjan

en la viabilidad de los descendientes. Este es el objetivo de la función de reparación de individuos, solo necesaria en estos casos.

2. Cruce por N puntos o multipunto (*multi-point crossover*): fue planteado por [28] como la generalización del cruce por un punto único. Consiste en fragmentar ambos cromosomas progenitores por N puntos aleatorios, obteniendo así N+1 fragmentos de cada uno. El descendiente se construye componiendo fragmentos alternados de cada padre de forma ordenada. Se ha comprobado experimentalmente que el valor N=2 ofrece mejores resultados que el corte por punto único. Sin embargo, no existe ningún estudio que confirme que valores superiores de N mejoren aun más los resultados. Este método muestra una tendencia mayor a fragmentar los cromosomas por las secciones centrales que por las zonas próximas a los extremos. Como en el caso anterior, es importante resolver los posibles conflictos derivados de la no viabilidad de los individuos descendientes, reparándolos adecuadamente.
3. Cruce uniforme o recombinación discreta (*uniform crossover*): puede considerarse como el caso extremo del cruce por N puntos en el sentido que cada gen es considerado un fragmento del cromosoma. El descendiente se construye permutando los genes de uno y otro progenitor con cierta probabilidad. En la mayoría de los casos se toma una probabilidad de 0,5 aunque algunos investigadores recomiendan una probabilidad algo inferior. Como en los anteriores tipos de cruce es necesario verificar si se precisa reparar individuos descendientes no viables, especialmente cuando la codificación es de longitud variable. La asignación del contenido para cada gen del cromosoma descendiente se hace de acuerdo a una máscara binaria de la misma longitud que los cromosomas progenitores, donde el valor “1” en la posición  $i$  significa asignar al gen  $i$  del descendiente el valor del mismo gen de un progenitor mientras que el valor “0” indica asignarle el contenido del gen  $i$  del otro progenitor. La Figura 2.6 muestra un ejemplo de cruce uniforme con los mismos cromosomas progenitores de la Figura 2.5. La máscara binaria tiene la misma longitud que los cromosomas y su contenido ha sido generado aplicando una probabilidad de 0,5. De modo que, en promedio, los descendientes reciben información de ambos progenitores al 50%. Sería posible obtener un segundo descendiente considerando la máscara complementaria a la empleada y con los mismos criterios.

Parent 1	A	B	C	D	E	F	G
Parent 2	1	2	3	4	5	6	7
Template	1	1	0	1	0	0	1
Descendent	A	B	3	D	5	6	G

**Figura 2.6:** Ejemplo de la operación de cruce uniforme en un algoritmo genético. El cromosoma descendiente se crea tomando los genes de uno u otro progenitor según indique la máscara binaria.

4. Cruce con permutación aleatoria asociada (*shuffle crossover*): es una técnica que se puede incorporar a los tres tipos de cruce anteriores con el fin de reducir la tendencia mayor que existe a fragmentar los cromosomas por las secciones centrales más que por los extremos.

Consiste en aplicar la misma permutación aleatoria a ambos progenitores antes de la operación de cruce. Después de la operación de cruce se debe aplicar la permutación inversa a los descendientes. De esta forma, las posiciones de los puntos de corte se distribuyen más uniformemente a lo largo de todo el individuo. La Figura 2.7 ofrece un ejemplo de permutación realizada con anterioridad al cruce de punto único de la Figura 2.5. El vector de permutación *shuffle* es (2 3 7 5 1 4 6) de la misma longitud (7 genes) que los cromosomas progenitores. El contenido del vector *shuffle* determina que el primer gen se desplaza a la segunda posición, el gen número 2 se lleva a la tercera posición, el tercer elemento del cromosoma se mueve a la última posición, el siguiente gen se lleva a la posición quinta, el gen número 5 se almacena en la primera posición y los dos últimos genes se guardan en las posiciones cuarta y sexta respectivamente. Tras esta permutación se realiza un cruce con corte entre los genes cuarto y quinto como en la Figura 2.5, aplicando después la permutación inversa sobre el cromosoma descendiente. El resultado de este tipo de cruce es el individuo (1 B C 4 E 6 G) con los genes heredados distribuidos de forma más uniforme que en el cruce simple por punto único, cuyo resultado fue (A B C D 5 6 7).

Parent 1	A	B	C	D	E	F	G
Parent 2	1	2	3	4	5	6	7
Shuffle	2	3	7	5	1	4	6
Parent 1 (*)	B	C	G	E	A	D	F
Parent 2 (*)	2	3	7	5	1	4	6
Descendent (*)	B	C	G	E	1	4	6
Descendent	1	B	C	4	E	6	G

**Figura 2.7:** Ejemplo, en un algoritmo genético, de la operación de cruce por punto único con permutación aleatoria asociada. Los cromosomas progenitores sufren la misma permutación aleatoria antes del cruce. Tras realizar el cruce, se aplica la permutación inversa al descendiente. (\*) Indica que el cromosoma está permutado.

5. Cruce de tipo *partial map crossover*: este operador se aplica en codificaciones basadas en el orden (permutaciones) donde el valor para un gen no puede repetirse en el mismo cromosoma. El ejemplo clásico para este tipo de codificación es el problema de empaquetamiento (*Packet Problem*), en el que cada paquete aparece representado una única vez en el contenido del cromosoma. Este tipo de cruce copia en el individuo descendiente parte de la información genética de uno de los dos progenitores, con su secuencia exacta y en la misma posición. El proceso de cruce es el siguiente: se aplica a uno de los progenitores N puntos de corte y se transfiere de manera íntegra al individuo descendiente fragmentos alternados del progenitor de forma similar al cruce multipunto. Para N=2 se copiarían en el descendiente los fragmentos primero y tercero del progenitor seleccionado. Los fragmentos restantes para el descendiente se completan con los valores de los genes no presentes en él y en el orden de aparición del segundo progenitor. La Figura 2.8 representa un ejemplo

de esta técnica con  $N=2$ , con los puntos de corte entre el segundo y tercer gen y entre el quinto y el sexto gen, respectivamente. Hemos elegido progenitores (A B C D E F G H) y el inverso (H G F E D C B A) con idea de simplificar el ejemplo. El cromosoma descendiente hereda los fragmentos extremos del primer progenitor resultando (A B - - - F G H). Para completar los tres huecos libres se eligen los genes del segundo progenitor por orden de aparición siempre y cuando no figuren ya en el individuo resultante. En este caso, los dos primeros genes del segundo progenitor (G F) no se consideran puesto que ya están incorporados en el descendiente, completando entonces con los siguientes (E D C).

Existen variantes de este tipo de cruce. Uno de ellos es conocido como cruce basado en el orden, donde una máscara binaria y aleatoria determina qué gen se transfiere de cada progenitor. Otra variante es el cruce cíclico, donde es el segundo progenitor el que determina qué genes del primero se transfieren al descendiente. Dado que el problema que nos ocupa en este trabajo de tesis no es de tipo permutación, no nos extenderemos más en los detalles de estos tipos de cruces.

Parent 1	A	B	C	D	E	F	G
Parent 2	G	F	E	D	C	B	A
Descendent	A	B	E	D	C	F	G

**Figura 2.8:** Ejemplo de cruce de tipo *partial map crossover* en un algoritmo genético. El cromosoma descendiente hereda fragmentos completos de uno de los dos progenitores, completando sus huecos con los genes todavía ausentes y en el orden de aparición en el segundo de los progenitores.

### 2.2.5 Operación de mutación

El operador de mutación se aplica generalmente sobre un porcentaje reducido de la población. Sin embargo, los efectos de la mutación son de gran importancia para la evolución del algoritmo. De hecho algunos algoritmos evolutivos, aunque no genéticos, utilizan la mutación como estrategia fundamental de búsqueda de soluciones. La idea que persigue esta operación es recrear las mutaciones genéticas producidas en las especies de la naturaleza y para cada generación, debidas a errores en la transferencia del ADN, constituyendo ésta una de las herramientas principales de la evolución natural de las especies. En los algoritmos genéticos, la mutación abarca varios objetivos:

- Exploración de nuevas zonas en el espacio de soluciones próximas a “soluciones de calidad” ya estudiadas.
- Procurar diversidad suficiente en la población evitando la convergencia prematura del algoritmo.
- En caso de producirse una convergencia prematura cayendo en algún óptimo local, poder liberar al algoritmo de ese valor de *fitness* máximo/mínimo local.

La operación de mutación se aplica tras el operador de cruce y solo a un porcentaje definido de la población. Dicho porcentaje suele variar entre el 1% y el 5% en codificación binaria, y

no mayor del 15% para codificación real. Un porcentaje excesivo de población mutada puede convertir la búsqueda de soluciones en una búsqueda prácticamente aleatoria. Mientras que un porcentaje demasiado pequeño no ayudaría al algoritmo a librarse de una posible convergencia prematura.

Para evitar que los mejores individuos de la población sea mutados con resultados de peor calidad suele conservarse una o varias copias de éstos evitando perderlos por una desafortunada mutación. Existen diferentes tipos de mutación dependiendo del tipo de codificación empleada. A continuación describimos brevemente los más representativos:

- En problemas con codificación binaria se aplica básicamente la mutación binaria. Este método es el más sencillo y popular. Una variante de ésta es la mutación binaria uniforme.
1. En la mutación binaria se genera una máscara binaria aleatoria de la misma longitud que el cromosoma que se va a mutar. Los genes del individuo que tienen la misma posición que los “1” en la máscara (*template*) son modificados, es decir, los “1” del individuo se convierten en “0” y viceversa. Los genes en las posiciones donde la máscara contiene “0” no sufren ninguna alteración. La Figura 2.9 muestra un ejemplo de este tipo de mutación para un cromosoma de 7 genes (1 1 0 1 0 1 1). La máscara (1 0 0 1 1 0 1) invierte los valores de los genes primero, cuarto, quinto y séptimo del cromosoma, resultando así el cromosoma (0 1 0 0 1 1 0).

Initial individual	1	1	0	1	0	1	1
Template	1	0	0	1	1	0	1
Mutation	0	1	0	0	1	1	0

**Figura 2.9:** Ejemplo de mutación binaria sobre un cromosoma. El contenido de los genes del cromosoma ubicados en las posiciones donde la máscara contiene “1” invierten su contenido, es decir los “1” se convierten en “0” y viceversa.

2. La mutación binaria uniforme, también denominada mutación binaria con probabilidad uniforme es una variación de la mutación binaria. La particularidad consiste en cómo se construye la máscara binaria *template*. Se genera un vector aleatorio de números reales en el intervalo [0-1] de la misma longitud que el cromosoma a mutar y se define un umbral de probabilidad o *threshold*. Las posiciones la máscara coinciden con un valor en el vector superior al umbral escriben un “1” en la máscara, y “0” el valor de vector es menor o igual al umbral. El procedimiento de mutación prosigue como en el caso de mutación binaria simple. La Figura 2.10 muestra un ejemplo de este tipo de mutación en el que se ha considerado un umbral de probabilidad de 0,4. El vector aleatorio de números reales junto con el umbral producen la máscara (0 0 1 1 0 1 1). Esta máscara aplicada sobre el mismo cromosoma de la Figura 2.9 (1 1 0 1 0 1 1), lo convierte en el nuevo individuo (1 1 1 0 0 0 0).
- Para codificaciones reales, enteras o finitas tenemos la mutación al borde y tres diferentes variantes de ésta: mutación de salto o uniforme, mutación no uniforme y mutación no uniforme modificada. Generalmente son aplicadas sobre uno o varios genes del cromosoma afectado por el operador de mutación. Cuando se realiza la operación sobre todos los genes del cromosoma (*multi-mutación*), el individuo resultante es mucho más diferente que el inicial. A continuación describimos brevemente los cuatro tipos de mutaciones.



Initial individual	1	1	0	1	0	1	1
Vector	0,2	0,4	0,8	0,6	0,1	0,9	0,7
Threshold=0,4							
Template	0	0	1	1	0	1	1
Mutation	1	1	1	0	0	0	0

**Figura 2.10:** Ejemplo de mutación binaria uniforme sobre un cromosoma de codificación binaria y con umbral de probabilidad  $threshold = 0,4$ . El vector de números reales aleatorios junto con el umbral de probabilidad generan la máscara binaria que se aplica sobre el cromosoma a mutar.

1. Mutación al borde o *boundary mutation*. La realidad es que los genes de un cromosoma presentan cierta dificultad para tomar los valores extremos dentro del rango o intervalo permitido, es decir, valores en los bordes de dicho intervalo. La mutación al borde trata de reducir esta tendencia. Una vez seleccionado el gen a modificar se le asigna un valor límite de su rango disponible  $[x_{min} - x_{Max}]$ : bien el mínimo  $x_{min}$  si la dirección es descendente, bien el máximo  $x_{Max}$  si la dirección en el intervalo es ascendente. En la Figura 2.11 se muestra un ejemplo de mutación al borde con dirección ascendente para el segundo gen del cromosoma. El contenido de este gen es 7,4 y el intervalo de posibles valores es  $[1-10]$ . Dado que la dirección es ascendente en dicho intervalo, el gen mutado pasa a tener el valor 10.

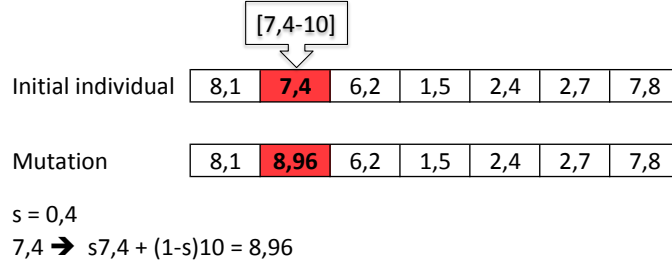
	[1-10]						
Initial individual	8,1	7,4	6,2	1,5	2,4	2,7	7,8
Mutation	8,1	10	6,2	1,5	2,4	2,7	7,8

**Figura 2.11:** Ejemplo de mutación al borde en un algoritmo genético con codificación no binaria. El gen número 2 es seleccionado al azar para que su contenido (7,4) sea modificado, eligiendo uno de los dos valores extremos del intervalo disponible  $[1-10]$ . En este caso se ha tomado el valor límite superior.

2. La mutación de salto (*step mutation*) es similar a la mutación al borde en el sentido de determinar el extremo superior o inferior del intervalo de valores para el gen que muta. El nuevo valor para este gen se elige de forma aleatoria en el rango delimitado por su valor actual y uno de sus valores límite. La Ecuación 2.5 expresa formalmente el cálculo para el nuevo valor en ambas direcciones siendo  $x_1$  el valor actual del gen y  $x_2$  su valor final para el mismo. El intervalo de valores disponible para dicho gen se representa como  $[x_{min} - x_{Max}]$  y  $s$  es una variable aleatoria que toma valores entre  $[0-1]$ . La Figura 2.12 muestra el mismo cromosoma del caso anterior pero afectado por una mutación de salto con dirección ascendente a  $x_{Max}$ . Para un valor de  $s = 0,4$ , el nuevo valor del gen es según la Ecuación 2.5 es  $x_2 = 8,96$ . Este tipo de mutación también se denomina como mutación uniforme para codificaciones no binarias porque la probabilidad de mutación permanece constante durante la evolución del algoritmo, al contrario que la mutación no uniforme en el que dicha probabilidad depende de la

generación que se esté ejecutando.

$$x_2 = \begin{cases} s \cdot x_1 + (1-s) \cdot x_{Max} & \text{dirección hacia el máximo valor} \\ (1-s) \cdot x_1 + s \cdot x_{min} & \text{dirección hacia el mínimo valor} \end{cases} \quad (2.5)$$



**Figura 2.12:** Ejemplo de mutación de salto en codificaciones no binarias. Tras elegir al azar el gen a mutar cuyo valor es 7,4 y considerando una dirección ascendente hacia el máximo valor del intervalo [1-10], el gen es modificado por valor aleatorio entre su valor inicial y el máximo posible.

3. Mutación no uniforme o *non-uniform mutation*. Como hemos mencionado arriba, el calificativo *no uniforme* hace referencia a que la probabilidad de mutación es variable, dependiendo del número de generaciones transcurridas. La probabilidad de mutación se va reduciendo conforme progresan las generaciones. En este aspecto es similar al proceso de recocimiento simulado o *simulated annealing*, donde el sistema se va *congelando* conforme progresa el algoritmo. La Ecuación 2.6 nos permite calcular la probabilidad de mutación  $p_i$  en la generación  $i$ , siempre variable dentro del rango  $[p_{min} - p_{max}]$ , donde  $p_{min}$  es su valor mínimo y  $p_{max}$  indica la probabilidad máxima de mutación. En esta ecuación,  $N_{max}$  se corresponde con el número máximo de generaciones que puede ejecutar el algoritmo.

$$p_i = p_{max} - \frac{i \cdot (p_{max} - p_{min})}{N_{max}} \quad (2.6)$$

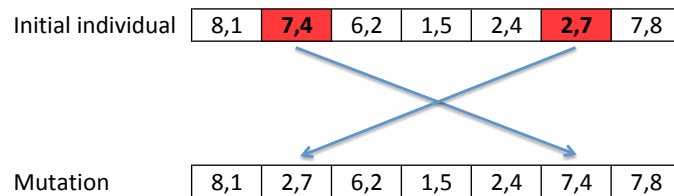
4. Mutación no uniforme modificada: es la opuesta de la mutación no uniforme en el sentido de que en este caso la probabilidad de mutación aumenta conforme progresa el algoritmo, en vez de reducirse como en la mutación no uniforme. Este método es más eficaz frente al problema de la convergencia prematura del algoritmo. En las primeras generaciones en general se estudian soluciones muy dispersas en el espacio de búsqueda puesto que existe suficiente diversidad y el proceso de mutación no es tan importante. Según progresa el algoritmo la diversidad tiende a reducirse, y éste va centrándose en áreas de soluciones más reducidas. La mutación no uniforme modificada permite saltos a nuevos espacios sin explorar evitando la convergencia en algún óptimo local. La Ecuación 2.7 expresa el cálculo de la probabilidad de mutación  $p_i$  creciente con el número de generaciones  $i$  recorridas.

$$p_i = p_{min} + \frac{i \cdot (p_{max} - p_{min})}{N_{max}} \quad (2.7)$$

- Para codificaciones basadas en el orden existen principalmente cuatro tipos de mutación, dependiendo del número de genes que se alteran y el modo en que se realiza: mutación de intercambio de dos genes cualesquiera, intercambio de dos genes adyacentes, intercambio

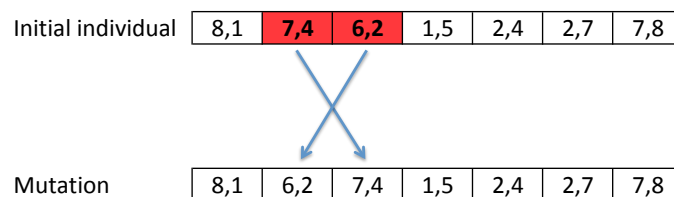
aleatorio de un conjunto de genes consecutivos o intercambio de un conjunto de genes consecutivos aplicando un desplazamiento definido a todos ellos. Para cada uno de estos cuatro tipos ofrecemos un ejemplo.

1. La mutación de intercambio o *swap mutation* consiste en intercambiar los valores de los genes elegidos de forma aleatoria. La Figura 2.13 muestra un ejemplo de mutación de intercambio de los genes segundo y penúltimo del cromosoma. Estos genes los hemos destacado en color rojo, siendo sus valores 7,4 y 2,7. La operación simplemente cambia la posición de estos genes. Una variante es la mutación de intercambio a tres o *3-swap mutation*, donde son tres los genes afectados por la mutación. Supongamos que las posiciones aleatorias para el cromosoma a mutar son las posiciones A, B y C. La mutación modifica los valores de los tres genes de manera que en la posición A se asigna el valor que hubiera en la posición B, y la posición B se actualiza con el contenido de la coordenada C y en C se almacena lo que hubiera en A antes de la mutación.



**Figura 2.13:** Ejemplo mutación de intercambio o *swap mutation* en un algoritmo genético con codificación basada en el orden. Dos genes con contenidos (7,4) y (2,7) intercambian sus posiciones.

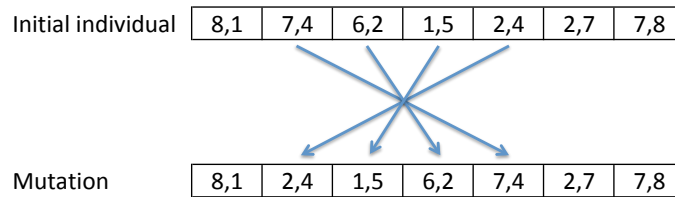
2. La mutación de intercambio con el adyacente (*adjoint swap mutation*) selecciona dos genes consecutivos en el cromosoma e invierte el orden de los mismos. En la Figura 2.14 se intercambian los genes segundo (7,4) y tercero (6,2) del cromosoma, también destacados en rojo.



**Figura 2.14:** Ejemplo de mutación de intercambio con el adyacente en codificaciones basadas en el orden para un algoritmo genético. El segundo gen (7,4) es seleccionado al azar y pasa a ocupar la posición de su adyacente (6,2). A su vez este último ocupa la posición del segundo.

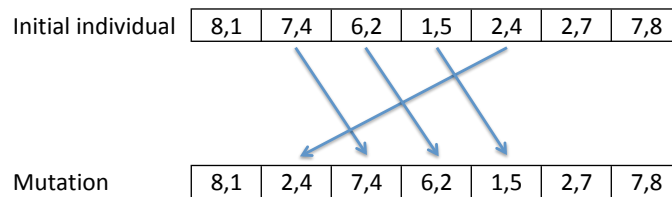
3. En la mutación de intercambio con varios consecutivos (*inversion mutation*) se eligen dos posiciones aleatorias en el cromosoma y se intercambian los valores de los genes entre dichas posiciones. Supongamos que las posiciones son  $i$  y  $j$ . La operación intercambia el valor del gen  $i$  con el  $j$  y viceversa, intercambia los genes  $(i+1)$  y  $(j-1)$ ,  $(i+2)$  con  $(j-2)$  y así sucesivamente hasta abarcar el intervalo  $[i - j]$  siendo los últimos genes intercambiados los  $(i+n)$  y  $(j-n)$ , con  $n = \frac{j-i+1}{2}$ . La Figura 2.15 muestra este tipo de mutación entre las posiciones segunda y quinta del cromosoma,

intercambiando el segundo (7,4) por el quinto (2,4) y el tercero (6,2) por el cuarto (1,5).



**Figura 2.15:** Ejemplo de mutación de inversión en un algoritmo genético, un conjunto de genes consecutivos intercambian sus valores dos a dos comenzando por los genes más alejados entre sí.

4. En la mutación con desplazamiento o *shift mutation* se determina un conjunto de genes consecutivos a alterar y una dirección de desplazamiento (hacia la derecha o hacia la izquierda). El proceso consiste en que cada gen del intervalo pasa a ocupar la posición inmediata hasta llegar al último gen afectado que se le hace corresponder con la primera posición del mismo grupo de genes. El efecto es que los valores de todos genes afectados se desplazan una posición en el cromosoma. En el ejemplo de la Figura 2.16 los genes alterados son como en el ejemplo anterior, el intervalo del segundo al quinto y la dirección de desplazamiento es hacia la derecha. El quinto gen (2,4) se lleva a la segunda posición y el resto se desplazan una posición hacia la derecha.



**Figura 2.16:** Ejemplo de mutación con desplazamiento en una algoritmo genético para codificaciones basadas en el orden. Definido un conjunto de genes consecutivos, el contenido de estos se desplaza una posición a la derecha en el cromosoma.

### 2.2.6 Búsqueda local

La búsqueda local es un método usado frecuentemente en la resolución de problemas de optimización combinatoria. Dada una solución  $s_0$ , se explora su vecindario buscando una solución  $s_1$  cercana a  $s_0$  y mejor que ésta. El vecindario de una solución está formado por el conjunto de soluciones a las que se llega con un pequeño movimiento o modificación en la solución. La solución  $s_1$  es mejor que  $s_0$  si su valor de función objetivo de la primera es más próximo al óptimo que el de la segunda (menor si se trata de minimizar la función de *fitness* o mayor si es el caso de maximizar). En cada iteración, la nueva solución puede ser la primera encontrada en el vecindario o bien la mejor solución dentro del vecindario. Esto último significa que una vez encontrado  $s_1$ , la búsqueda local se realiza sobre el vecindario de  $s_1$  y así sucesivamente hasta que no exista en el vecindario de la nueva solución ninguna otra con mejor valor de la función objetivo. Debido precisamente a este método de búsqueda, el principal problema de esta técnica es la gran facilidad con la que el algoritmo queda atrapado en óptimos locales y rara vez descubre el óptimo global. Para evitar este desenlace existen varias posibilidades:

- Ampliación del tamaño del vecindario.
- Limitar el número de iteraciones de búsqueda.
- Repetir el algoritmo de búsqueda con diferentes soluciones de partida  $s_0$ .

Ninguno por separado ha generado resultados satisfactorios. Lo que genera los mejores resultados es combinar la técnica de búsqueda local con otro tipo de mecanismos en lo que se denomina algoritmos híbridos.

## 2.3 Introducción al algoritmo PSO

El algoritmo de Optimización por Cúmulos de Partículas o *Particle Swarm Optimization* (PSO) es una técnica metaheurística bio-inspirada, basada en el comportamiento social de poblaciones grupales como bandadas de aves o bancos de peces, comportamiento social para volar juntas en el primer caso y el movimiento colectivo bajo el agua en el segundo. El PSO fue originalmente desarrollado por dos investigadores: el psicólogo-sociólogo James Kennedy y el ingeniero electrónico Russell Eberhart en 1995, basándose en el enfoque de la *metáfora social* [63], que describe a este hecho y que se puede resumir de la siguiente forma: los individuos que conviven en una sociedad tienen una opinión que es parte de un conjunto de creencias (el espacio de búsqueda) compartido por todos los posibles individuos. Además cada individuo puede modificar su propia opinión basándose en tres factores:

- Su conocimiento sobre el entorno (aquí queda incluido la función objetivo o función de *fitness*).
- Su conocimiento histórico o experiencia personal anterior, es decir, su memoria.
- El conocimiento histórico o experiencias anteriores de los individuos en su vecindario. Lo cual tiene se asocia a la relación social entre individuos.

Según transcurre el tiempo en la naturaleza, los individuos del grupo modifican su posición adaptando sus movimientos a la de los seres de su especie con más éxito en su entorno, relacionando estrechamente el comportamiento de todos entre sí. El algoritmo PSO simula el comportamiento del vuelo de las bandadas de aves (ver imagen de la izquierda de la Figura 2.17) o bancos de peces (imagen derecha de la Figura 2.17) en busca de comida. La estrategia más eficaz es seguir al ave que está más cerca de la comida. El algoritmo modela cada ave como una partícula con una solución en el espacio de búsqueda que está siempre en continuo movimiento y nunca se elimina o muere. El PSO es un sistema *multiagente*, es decir, las partículas son agentes simples que se mueven por el espacio de búsqueda, que guardan y en ocasiones comunican la mejor solución que han encontrado.

### 2.3.1 Descripción de un algoritmo PSO

El algoritmo PSO consiste en un proceso iterativo y estocástico que opera sobre un cúmulo de partículas. La posición de cada partícula representa una solución potencial al problema de optimización a resolver. Para cada partícula  $i$  se dispone de cierta información, distribuida en tres vectores y dos valores de *fitness*:

- El vector  $x_i = \langle x_{i1}, x_{i2}, \dots, x_{in} \rangle$  almacena la posición actual o localización de la partícula  $i$  en todo el espacio de búsqueda.



**Figura 2.17:** Ejemplos de grupos de animales o *swarms* en la naturaleza (aves en la parte izquierda y peces en la imagen derecha de la figura). Cada animal decide su movimiento en base a su propio conocimiento y a su experiencia dentro del grupo.

- El vector  $pBest_i = \langle p_{i1}, p_{i2}, \dots, p_{in} \rangle$  guarda la posición de la mejor solución encontrada por la partícula  $i$  durante toda su existencia en la ejecución del algoritmo.
- El vector de velocidad  $v_i = \langle v_{i1}, v_{i2}, \dots, v_{in} \rangle$  registra la dirección según la cual se moverá la partícula  $i$  en la siguiente iteración del proceso.
- El valor de *fitness* actual  $\mathcal{F}_i$  de la partícula  $i$ , obtenido a partir de la posición de la misma y registrada en el vector  $x_i$ .
- El mejor valor de *fitness*  $\mathcal{F}_{pBest_i}$  que la partícula  $i$  adquirió en algún momento durante la ejecución del método.  $\mathcal{F}_{pBest_i}$  se calcula a partir del vector  $pBest_i$ .

Se entiende por *swarm*, cúmulo de partículas o simplemente *swarm*, el conjunto de individuos que se desplazan a través del espacio de soluciones con una velocidad determinada. Para cada elemento del *swarm* se genera de forma aleatoria tanto su posición como su velocidad iniciales. A partir de ese momento cada individuo decidirá su posición y velocidad siguientes de acuerdo al comportamiento social del cúmulo. Las velocidades están sujetas generalmente a un intervalo del tipo  $[-v_{max}, v_{max}]$ , donde  $v_{max}$  representa la velocidad máxima que puede tomar la partícula en un movimiento.

Una vez identificada la nueva posición de la partícula  $i$ , se evalúa su posición calculando  $\mathcal{F}_i$ . En el caso de ser éste el mejor valor de *fitness* encontrado durante la existencia de la partícula  $i$ , se actualizan también el valor de los parámetros  $pBest_i$  (la mejor posición de la partícula  $i$ ) y  $\mathcal{F}_{pBest_i}$  (el valor de la función de evaluación de  $i$  en la posición  $pBest_i$ ).

Las Ecuaciones 2.8 y 2.9 representan el modelo matemático del algoritmo PSO. Se trata de un proceso iterativo, en el que a partir de los parámetros de la partícula  $i$  en la iteración  $k$  se obtienen los valores de los mismos para la iteración inmediata posterior ( $k+1$ ). La Ecuación 2.8 permite conocer la nueva posición de la partícula  $i$  a partir de la posición actual  $x_i^k$  y el vector velocidad para la nueva iteración  $v_i^{k+1}$ .

La Ecuación 2.9 corresponde a la versión clásica de [63]. Conocidas la posición  $x_i^k$  y la velocidad  $v_i^k$  de la partícula  $i$  en la iteración  $k$ , el vector velocidad  $v_i^{k+1}$  en la iteración siguiente ( $k+1$ ) resulta de la combinación lineal de tres componentes: la propia velocidad  $v_i^k$  de la partícula  $i$  en la iteración  $k$ , un componente *cognitivo* y un componente *social* que a continuación describimos. El hecho de trabajar con la velocidad en dos iteraciones sucesivas caracteriza al método como

un algoritmo *con memoria*.

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2.8)$$

$$v_i^{k+1} = v_i^k + \varphi_1 \cdot rand_1 \cdot (pBest_i - x_i^k) + \varphi_2 \cdot rand_2 \cdot (g_i - x_i^k) \quad (2.9)$$

La parte cognitiva se modela mediante el sumando  $\varphi_1 \cdot rand_1 \cdot (pBest_i - x_i^k)$  y representa la distancia entre la posición actual  $x_i^k$  y la mejor posición  $pBest_i$  para la partícula  $i$  hasta el momento. De forma cualitativa, indica la influencia de la experiencia de la partícula a lo largo de su existencia. Los términos  $\varphi_1$  y  $rand_1$  permiten asignar un mayor o menor peso a dicha influencia.

La componente *social* se define a través del término  $\varphi_2 \cdot rand_2 \cdot (g_i - x_i^k)$ . El significado físico es la distancia entre la posición actual  $x_i^k$  de la partícula  $i$  y la mejor posición  $g_i$  de su vecindario. El vecindario puede considerarse como el *swarm* completo o solo las más próximas a la partícula. Este término representa la influencia que ejerce un conjunto de partículas vecinas sobre la partícula estudiada. De forma análoga a la componente cognitiva, los términos  $\varphi_2$  y  $rand_2$  permiten asignar un mayor o menor peso a la influencia social. A continuación enumeramos algunas de las variables protagonistas en el modelo.

- $v_i^k$  es la velocidad de la partícula  $i$  en la iteración  $k$ , siempre dentro de un intervalo establecido  $[-v_{max}, v_{max}]$ .
- $\varphi_1, \varphi_2$  son factores de aceleración, también llamados ratios de aprendizaje para las componentes cognitiva y social respectivamente.
- $rand_1, rand_2$  representan variables aleatorias en el intervalo  $[0,1]$ . El algoritmo PSO adquiere con ellas un carácter estocástico.
- $x_i^k$  es la posición actual en el espacio de soluciones de la partícula  $i$  en la iteración  $k$ .
- $pBest_i$  es la mejor posición encontrada en el espacio de búsqueda para la partícula  $i$  hasta el momento.
- Por último,  $g_i$  representa la posición de la partícula con el mejor valor de *fitness*  $\mathcal{F}_{pBest}$  en su vecindario. Éste puede ser un vecindario global, el *swarm* completo, o un vecindario local si considera un conjunto reducido de partículas vecinas la  $i$ .

En cuanto al valor de los ratios de aprendizaje  $\varphi_1$  y  $\varphi_2$  y la relación entre ellos, existen diversos estudios. En [61] se trabajó con el modelo cognitivo puro ( $\varphi_1 > 0$  y  $\varphi_2 = 0$ ), el modelo social puro ( $\varphi_1 = 0$  y  $\varphi_2 > 0$ ) y el modelo completo ( $\varphi_1, \varphi_2 > 0$ ). Por otro lado, [34, 56] recomiendan asignar el mismo peso a ambos factores, estableciendo los valores  $\varphi_1 = \varphi_2 = 2$ .

Respecto al tratamiento de vecindario global y/o local en el algoritmo, ambos casos fueron estudiados en [32, 63], respectivamente. Un entorno global converge habitualmente más rápido que otro local puesto que la visibilidad de cada partícula es total. De esta manera se estimula la intensificación sobre todo el espacio de búsqueda de soluciones. Por contra, el vecindario global tiene un riesgo ligeramente mayor para quedar atrapado en óptimos locales. Por otro lado, un vecindario local converge en general de forma lenta, favoreciendo la diversificación en el espacio de búsqueda. El modelo de vecindario local se ha implementado con diversas topologías: centralizada en [8], topología de anillo (*wheel* o también conocida por topología *star*) en [62] y topología de *Von Neumann* (toroidal o cuadrada) en [64, 65]. Para la resolución del problema

MNDP que nos ocupa, hemos considerado un vecindario local con topología centralizada y con tamaño de 3 partículas, ver Capítulo 4.

La velocidad máxima  $v_{max}$  sirve para controlar el modo de exploración dentro del espacio de búsqueda. Un valor elevado de  $v_{max}$  favorece la exploración global, mientras que un valor reducido de  $v_{max}$  facilita la exploración local. Es importante encontrar un valor adecuado que mantenga el equilibrio entre ambas opciones. Si el valor de  $v_{max}$  es demasiado alto las partículas pueden sobrepasar e ignorar continuamente la zona que contiene la solución óptima. Por el contrario, si  $v_{max}$  toma valores extremadamente bajos las partículas explorarán el espacio de soluciones muy lentamente y podrían quedar atrapadas alrededor de óptimos locales, incapaces de librarse de su atracción. El efecto de  $v_{max}$  está fuertemente vinculado a la naturaleza del problema a resolver y resulta difícil de ajustar. Con el objetivo de reducir el efecto de  $v_{max}$  y perfeccionar la búsqueda de soluciones se desarrollaron versiones con dos nuevos parámetros: *inertia weight*  $\omega$  y factor de constricción  $\chi$ .

En [95, 96] se introdujo el parámetro *inertia weight*  $\omega$  tratando de obtener un mejor balance entre exploración global (diversificación) y explotación local (intensificación) reduciendo la influencia de la velocidad  $v_i^k$  de la partícula en el cálculo de cada iteración, según se muestra en la Ecuación 2.10.

$$v_i^{k+1} = \omega \cdot v_i^k + \varphi_1 \cdot rand_1 \cdot (pBest_i - x_i^k) + \varphi_2 \cdot rand_2 \cdot (g_i - x_i^k) \quad (2.10)$$

En estos estudios, el valor de *inertia weight* inicial era constante y su efecto era similar al de  $v_{max}$ : valores elevados favorecían la búsqueda global mientras que valores reducidos facilitaban la búsqueda local. En [97] se comprobó experimentalmente que valores de *inertia weight* que decrecientes de forma lineal mejoraban la eficiencia del PSO. De esta forma se mantenía el algoritmo en estados de exploración global durante las primeras iteraciones y posteriormente intensificaba la búsqueda en entornos locales favorables con valores de velocidad menores, reduciendo el parámetro *inertia weight*.

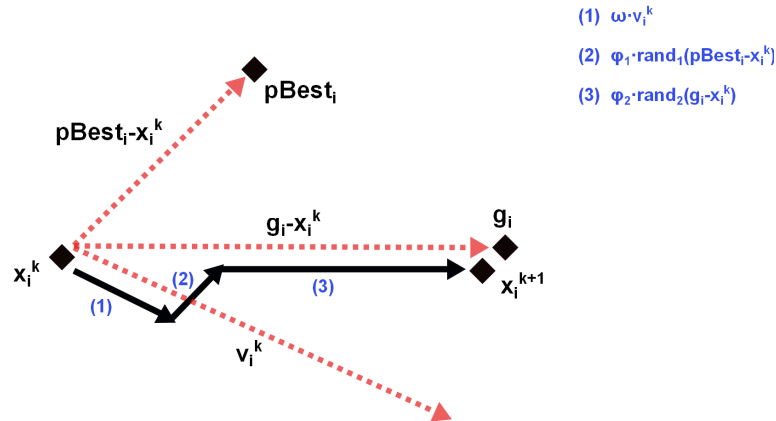
En [18, 20] se presentan modelos similares introduciendo el parámetro factor de constricción  $\chi$ . En realidad, un algoritmo PSO con factor de constricción es un caso especial del algoritmo con *inertia weight*. En un estudio comparativo de ambos en [33] determinaron los valores de los parámetros analíticamente llegando a esta conclusión. En nuestro trabajo utilizamos el factor *inertia weight* en vez del factor de constricción, de modo que no nos vamos a extender en la explicación de sus características.

La Figura 2.18 representa gráficamente las Ecuaciones 2.8 y 2.10, donde según la Ecuación 2.8, la posición  $x_i^{k+1}$  de la partícula  $i$  en la iteración  $(k+1)$  es obtenida a partir de su posición  $x_i^k$  en la iteración inmediata anterior  $k$  y el vector velocidad  $v_i^{k+1}$  de la partícula en la iteración siguiente  $(k+1)$ . El vector velocidad se desglosa a su vez en las tres componentes reflejadas en la Ecuación 2.10. En la figura, las posiciones aparecen como puntos negros de forma cuadrada: tanto las posiciones de la partícula  $i$  como la mejor posición de  $i$  durante la ejecución del algoritmo  $pBest_i$  y la posición más favorable entre sus vecinos  $g_i$ . Las flechas en color rojo con trazo discontinuo indican la dirección de las tres componentes de la Ecuación 2.10, mientras que los vectores en línea continua negra delimita el módulo y dirección de dichas componentes. En la leyenda de la figura se indican la magnitud de las mismas antes de realizar la suma vectorial de la Ecuación 2.10:

- (1) Vector velocidad  $v_i^k$  de la partícula  $i$  en la iteración  $k$ . Aparece multiplicado por el factor *inertia weight*  $\omega$  que controla el efecto de la velocidad de la partícula según la Ecuación 2.10.



- (2) Factor *cognitivo*: proporcional a la diferencia vectorial de la posición de la partícula y la mejor posición de la misma hallada hasta el momento.
- (3) Factor *social*: proporcional a la diferencia vectorial de la posición de la partícula y la mejor posición alcanzada en su vecindario.



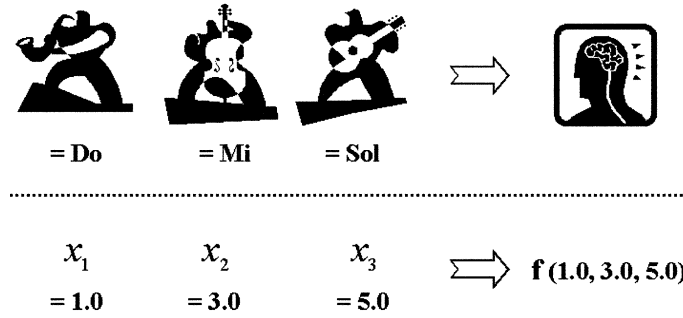
**Figura 2.18:** Movimiento desde la iteración  $k$  a la iteración  $(k+1)$  de la partícula  $i$  en el espacio de búsqueda de soluciones. A partir de su posición en la iteración  $k$  se determina la posición en  $(k+1)$  bajo el modelo de las Ecuaciones 2.8 y 2.10.

## 2.4 Introducción al algoritmo Harmony Search

El algoritmo *Harmony Search* (HS) o algoritmo de búsqueda armónica es un algoritmo heurístico que surgió a comienzos de siglo a partir del trabajo del investigador Z. W. Geem [45]. Se trata de un algoritmo inspirado en la improvisación de una orquesta de jazz trabajando en la composición de una nueva pieza musical, de la cual se desea una armonía perfecta. Se puede trazar una correspondencia entre este ejemplo con la resolución de un problema de optimización, donde cada músico representa una variable de decisión de la función objetivo y cada nota musical se asocia a un valor posible para las diferentes variables de decisión. El grado de musicalidad de un acorde se corresponde con el valor de *fitness* de una solución al problema de optimización. La perfecta armonía de la melodía se asocia con el óptimo global del problema [42]. La Figura 2.19, recogida de [68], representa esta analogía, donde cada uno de los tres instrumentos musicales (saxofón, contrabajo y guitarra) se asocian a una variable  $(x_1, x_2, x_3)$  de la función objetivo. El acorde (Do, Mi, Sol) resulta más o menos armonioso para el público de la misma forma que el valor de la función objetivo  $f(x_1, x_2, x_3) = f(1, 3, 5)$  se aproxima en mayor o menor medida a su valor óptimo.

El algoritmo HS es relativamente sencillo de implementar, además consigue bastante buenos resultados. Ha sido empleado en aplicaciones muy diversas: problemas de optimización en ingeniería [68, 71, 76], redes de comunicación [39, 46], minería de datos [40], redes de distribución de aguas [43, 73], modelos de flujos de aguas subterráneas, sistemas de eficiencia energética y ahorro de energía, aplicaciones arquitectónicas, problemas de tráfico de vehículos [47] y problemas matemáticos como *sudokus* [44].

En la composición de una pieza musical, cada músico de la orquesta tiene tres formas distintas de actuar para la elección de qué nota tocar en cada momento o acorde, a la vez que sus



**Figura 2.19:** Analogía entre una orquesta de jazz compuesta por tres músicos probando un nuevo acorde (Do, Mi, Sol) y la evaluación de la solución ( $f(x_1, x_2, x_3) = f(1, 3, 5)$ ) en un problema de optimización. Figura tomada de [68].

compañeros tocan sus respectivos instrumentos:

1. Tocar una nota conocida por el músico previamente, que además sabe que tuvo buen resultado en acordes anteriores.
2. Tocar una nota conocida como en el caso anterior, pero con alguna variación intencionada.
3. Improvisar completamente la nota en el acorde correspondiente.

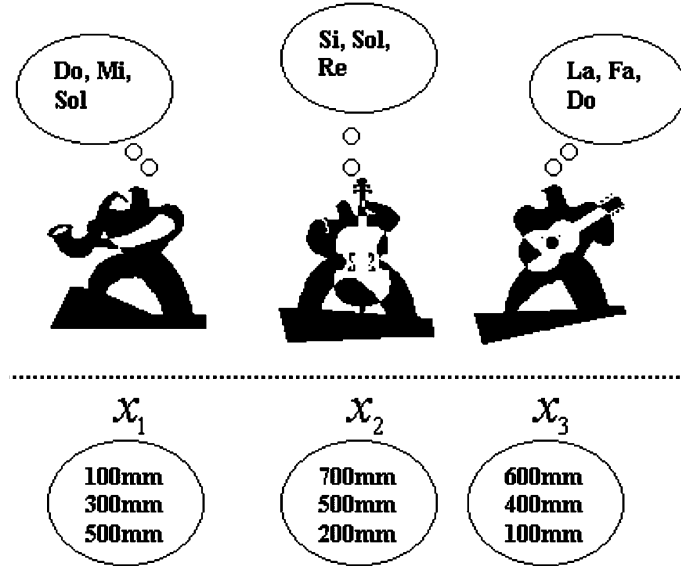
El algoritmo HS define una memoria de armonías o *harmony memory* (HM) que almacena un conjunto limitado de soluciones al problema de optimización que ofrecen los mejores resultados hallados hasta ese momento. El número de soluciones registradas en la HM o tamaño de la memoria se conoce como *harmony memory size* (HMS). En el símil de la orquesta de jazz, la HM corresponde al conocimiento de acordes musicales aprendidos por los músicos. En un algoritmo genético, la HM es el conjunto de las mejores soluciones halladas.

Cada una de las tres opciones descritas anteriormente para el ejemplo de la orquesta se corresponde con una herramienta diferente en el algoritmo HS [45]:

1. Utilización del conocimiento de la HM *memory consideration*.
2. Variaciones sobre información extraída de la HM, lo cual se denomina en jerga musical como ajuste de tono o *pitch adjustment*.
3. Aleatoriedad o *randomization*.

La Figura 2.20, recogida también de [68], representa gráficamente la analogía del uso de la HM en una orquesta de jazz y en la función objetivo de un problema de optimización de energía. Como en la figura anterior, la orquesta se compone por tres músicos (saxofón, contrabajo, guitarra), a la vez que la función objetivo se define mediante tres variables de decisión ( $x_1, x_2, x_3$ ). Los músicos disponen en la HM de tres acordes ( $x^1, x^2, x^3$ ), de la misma manera que la función objetivo cuenta con tres conjuntos de valores que generan las tres mejores soluciones encontradas hasta el momento. El tamaño de la memoria o HMS es, por lo tanto, de tres soluciones. La Tabla 2.2 estructura el contenido de la HM en forma matricial, donde las filas representan los acordes (soluciones al problema de optimización) y las columnas indican las líneas melódicas para cada instrumento (conjunto de valores almacenados en la memoria para cada variable de la función objetivo). El elemento  $x_j^i$  se localiza en la fila  $i$  y en columna  $j$ , su contenido es precisamente la

nota del acorde  $i$  para el instrumento musical  $j$ . Por ejemplo, la fila  $x^1 = (Do, Si, La)$  establece que el saxofonista toque la nota  $Do$ , el contrabajo la nota  $Si$ , a la vez que el guitarrista toque la nota  $La$ .



**Figura 2.20:** Ejemplo del uso de la *harmony memory* (HM). Analogía entre una orquesta de jazz con tres acordes aprendidos y el conjunto de tres soluciones encontradas para un problema de optimización de energía. Figura tomada de [68].

**Tabla 2.2:** Contenido de la *harmony memory* de la Figura 2.20 en forma matricial. Cada fila representa un acorde musical (solución del problema de optimización) y cada columna se corresponde con un instrumento musical (variable de decisión de la función objetivo).

	Saxofón	Contrabajo	Guitarra
	$x_1$	$x_2$	$x_3$
$x^1$	Do	Si	La
$x^2$	Mi	Sol	Fa
$x^3$	Sol	Re	Do

La operación definida arriba como *memory consideration* consiste en generar un nuevo acorde a partir de notas registradas en la HM, aun no correspondiéndose todas ellas a un mismo acorde o fila de la matriz. Para ello, se elige un elemento de cada columna (instrumento musical *versus* variable de decisión) variando las filas bajo ciertas restricciones probabilísticas que describimos a continuación. La herramienta de *pitch adjustment* parte del acorde generado mediante *memory consideration* y modifica intencionadamente algunas notas del mismo, también de forma controlada con otros parámetros que definimos en los siguientes párrafos. La última acción, denominada *randomization*, no hace uso de la memoria y simplemente genera un nuevo acorde con todas las notas aleatorias.

El algoritmo HS consiste en realizar un número determinado de iteraciones hasta cumplir la condición de parada del algoritmo. En cada iteración se crea un nuevo acorde  $x'$  o solución, la cual es evaluada mediante la función objetivo. En caso de tener mejor calidad que alguna de las registradas en la memoria,  $x'$  pasa a ocupar el lugar de la peor de las soluciones de la HM.

En caso de que  $x'$  no mejore a ninguna de las soluciones de la memoria, ésta última permanece invariable hasta la siguiente iteración.

Para la generación de un nuevo acorde  $x'$  mediante la operación de *memory consideration* se define la probabilidad *harmony memory considering rate* o HMCR. A cada instrumento  $i$  se le asigna una nota  $x'_i$  del acorde  $x'$ , que es tomada de la columna  $i$  de la matriz HM con probabilidad HMCR o bien es seleccionada al azar entre el rango de valores disponibles  $X_i$  con probabilidad  $(1-\text{HMCR})$ . En el primer caso se asume que hay equiprobabilidad entre los elementos del vector columna  $\{x_i^1, \dots, x_i^{\text{HMS}}\}$ , donde HMS es el número de filas de la matriz (soluciones almacenadas en la HM, o tamaño de la memoria). Este es el caso de la operación de *memory consideration* propiamente dicha. El segundo caso, con probabilidad  $(1-\text{HMCR})$ , responde a la operación de *randomization*. Aunque la operación de *randomization* se enumeró en último lugar, en la práctica se realiza con probabilidad complementaria a la de *memory consideration*. La Ecuación 2.11 recoge ambas acciones, la primera línea se corresponde con *memory consideration* y la segunda con *randomization*. Según esta ecuación, la nueva nota  $x'_i$  para el instrumento  $i$  es una aleatoria registrada previamente en la memoria para dicho instrumento con probabilidad HMCR, o bien toma cualquier otro valor en el rango  $X_i$  con la probabilidad complementaria. Generalmente el valor de HMCR es elevado, en torno a 0,95 [68]. El 5% restante, correspondiente a  $(1-\text{HMCR})$ , tiene un efecto similar al de la mutación en un algoritmo genético. Algunos autores recomiendan no mantener constante el valor de HMCR, sino variarlo durante la evolución del propio algoritmo, aunque siempre asignándole valores altos, en [108] se recomienda el intervalo  $\text{HMCR} \in [0,7 - 0,95]$ .

$$x'_i \leftarrow \begin{cases} x'_i \in \{x_i^1, x_i^2, \dots, x_i^{\text{HMS}}\} & \text{con probabilidad HMCR} \\ x'_i \in X_i & \text{con probabilidad } (1 - \text{HMCR}) \end{cases} \quad (2.11)$$

La herramienta de *pitch adjustment* o ajuste de tono del algoritmo HS también recuerda en cierto modo a la operación de mutación de un algoritmo genético, que persigue explorar nuevos espacios del conjunto de soluciones. En la analogía de la orquesta de jazz, el ajuste de tono consiste en variaciones ligeras de frecuencias para algunos o todos los instrumentos. En el HS esta acción queda definida mediante dos parámetros: el *pitch adjustment rate* o PAR, que marca la probabilidad de ajuste de una nota  $x'_i$  y el *pitch band width* ( $bw$ ) o ancho de banda de ajuste que se aplica sobre el rango de valores disponibles para dicha nota  $X_i$ . Para cada nota  $x'_i$  del nuevo acorde  $x'$  se aplica un ajuste de tono con probabilidad PAR. Esto implica que la nota permanece invariable con probabilidad  $(1-\text{PAR})$ . El modo de ajuste de las notas puede ser lineal o no lineal. Generalmente se utiliza un ajuste lineal como el descrito en la Ecuación 2.12. En ella,  $\epsilon$  es un número aleatorio que responde a la distribución uniforme  $u(-1, 1)$ . Como ejemplo supongamos que el la nota  $x'_i$  que debe ajustarse tiene el valor  $k^{\text{th}}$  en el rango  $X_i$ . Si el producto  $bw \times \epsilon = 4$  significa que el nuevo valor para  $x'_i$  será el ubicado en la posición  $(k + 4)^{\text{th}}$  del rango  $X_i$ .

$$x'_i \leftarrow x'_i + bw \cdot \epsilon \quad (2.12)$$

Una probabilidad PAR junto con un ancho de banda  $bw$  ambos reducidos consiguen que el algoritmo converja lentamente puesto que no provocan saltos notables en el espacio de soluciones. En el extremo opuesto se encuentran valores elevados tanto para el PAR como para el  $bw$ , que hacen que el algoritmo realice una búsqueda casi aleatoria moviéndose continuamente por todo el espacio de búsqueda. En la práctica [108] se utilizan valores de  $\text{PAR} \in [0,1 - 0,5]$ . Queremos hacer notar que la probabilidad real de la operación de *pitch adjustment* para un nuevo acorde

es  $HMCR \times PAR$ , puesto que esta acción se realiza si y solo si se ha aplicado con anterioridad *memory consideration*. A modo de resumen, indicamos las probabilidades con las que se realizan cada una de las tres operaciones del algoritmo HS:

- Utilización del conocimiento de la HM con probabilidad  $HMCR$ .
  - Acción de *memory consideration* con probabilidad  $HMCR \times (1 - PAR)$ .
  - Acción de *pitch adjustment* con probabilidad  $HMCR \times PAR$ .
- Operación de aleatoriedad o *randomization* con probabilidad  $(1-HMCR)$ .

## 2.5 Aplicaciones de técnicas de heurísticas en problemas de localización de estaciones base en redes de telecomunicación

El problema de optimización para el diseño de redes de telecomunicación se ha estudiado extensamente en la literatura. En la gran mayoría de los casos se considera el parámetro “coste económico” como variable protagonista de la función objetivo. En algunos casos se incluyen además otras variables que penalizan dicha función, como parámetros derivados del tráfico bloqueado, la capacidad no asignada o el tráfico demandado. El coste económico es calculado fundamentalmente a partir del equipamiento de la red planteada. Puede añadirse también el coste asociado a la interconexión de nodos u otros elementos decisivos de la red. En otros casos como [1] no se analiza coste económico, sino que el objetivo es minimizar el rango dinámico de canal de red.

Hasta la fecha no hemos encontrado estudios que incluyan en su función de evaluación la variable de radiación electromagnética generada por el funcionamiento propio de la red de telecomunicación.

Entre las referencias que aplican algoritmos de tipo evolutivo al problema de localización de estaciones base se encuentra [25], que analiza desde un punto de vista económico la planificación de redes de fibra óptica e implementa un algoritmo genético para resolver el problema de diseño topológico de redes de alta velocidad. En [109] se describe un algoritmo evolutivo híbrido, mientras que en [90] se utilizan los algoritmos genéticos híbridos para la asignación de terminales bajo restricciones de coste y capacidad.

Otros estudios comparan los algoritmos genéticos con la búsqueda tabú, aplicados ambos sobre el problema de asignación de elementos o diseño red. Es el caso de [17], que realiza el diseño de redes minimizando los costes de nodos y enlaces. [67] aplica también ambos métodos sobre el problema de localización de estaciones base. En [84] se definen las restricciones de coste económico y capacidad para el problema de asignación de *switches*, comparando los resultados de tres métodos diferentes: búsqueda tabú, *simulated annealing* y algoritmo genético. [98] aplica la estrategia de *simulated annealing* para el diseño de redes donde se desea optimizar las variables de coste de la red, capacidad de los equipos y demanda de sus clientes.

En parte de la literatura se implementan diversos métodos heurísticos híbridos sobre este tipo de problemas [87]. En [72] se desarrolla un algoritmo heurístico para el diseño de redes de acceso basado en el método de relajación del problema de optimización (*Lagrangian relaxation*) que minimiza los costes de interconexión entre nodos.

En [83] se resuelve el problema de diseño de redes móviles con una heurística de búsqueda local bajo restricciones de coste y capacidad de tráfico desglosados para los diferentes componentes de la red: MSC (*mobile switching center*), BSC (*base station controllers*) y enlaces. En [86] se trata un problema similar añadiendo a la función objetivo un coste derivado del número de *handovers* realizados.

En [14] se presenta un algoritmo heurístico para redes UMTS contemplando el coste debido a la inserción de nodos, interfaces y enlaces, además del coste derivado por tráfico bloqueado. Finalmente, en [60] se aplican técnicas metaheurísticas en el coste de redes UMTS.

## Capítulo 3

# Planteamiento matemático del problema MNDP

En este capítulo exponemos la formulación matemática del problema MNDP (*Mobile Network Deployment Problem*) o despliegue de redes móviles con control de contaminación electromagnética. Consideremos un área metropolitana  $A$ , del cual disponemos de una serie de medidas de campo eléctrico ( $V/m$ ) referente a un conjunto de puntos  $\mathcal{O} = \{\mathbf{o}_i\}$ ,  $i \in \{1, \dots, O\}$ . Estos puntos quedan definidos por sus coordenadas  $(x_i^o, y_i^o) \in A$ . De forma teórica, la superficie  $A$  se dividiría en una cuadrícula de manera que cada celda estuviese representada por un único punto. Y la agrupación de puntos representantes de las celdas formarían el conjunto  $\mathcal{O}$ . Sin embargo en la práctica, algunas cuadrículas son de escaso interés para el servicio de telecomunicación y por lo tanto dejan de estar representadas en el conjunto  $\mathcal{O}$ . Otras cuadrículas son de difícil acceso para la medición de campo eléctrico por diversos motivos, incluso a veces resulta inviable llevar a cabo la medida, por ejemplo, cuando se trata de un lago. Por estas razones el conjunto de puntos  $\mathcal{O}$  no está uniformemente distribuido. Consideremos además los siguientes puntos que ayudan a definir el problema:

- Definimos  $\mathcal{P}$  como el conjunto de puntos donde es posible instalar las estaciones base. Formalmente,  $\mathcal{P} = \{\mathbf{p}_i\}$ ,  $i \in \{1, \dots, P\}$ . Cada punto está definido por sus coordenadas  $(x_i, y_i) \in A$ .
- Consideramos un conjunto de tipos de estaciones base definido como  $\mathcal{T} = \{\mathbf{t}_i\}$ ,  $i \in \{1, \dots, T\}$ . El tipo de una estación base depende del valor asignado en tres parámetros: frecuencia de trabajo de la estación, potencia de transmisión y coste económico asociado.
- Establecemos el conjunto de puntos  $\mathcal{P}'$  en la zona de estudio  $A$  donde se desea ofrecer el servicio de telecomunicación. De esta forma definimos  $\mathcal{P}' = \{\mathbf{p}'_i\}$  con  $i \in \{1, \dots, M\}$ , donde cada punto queda unívocamente definido por sus coordenadas  $(x'_i, y'_i)$ .
- Por último,  $\mathcal{C}(D)$  es la función de coste que indica el coste económico de las estaciones base que componen la red de telecomunicación. Esta función de coste depende del número de equipos y del tipo de los mismos, de modo que  $\mathcal{C}(D) : \mathcal{P} \times \mathcal{T} \rightarrow \mathbb{R}$ .

Una solución del problema MNDP consiste en un despliegue de red  $D = \{(\mathbf{p}_i, \mathbf{t}_i)\}$ ,  $i \in \{1, N\}$ , en el que cada estación base  $i$  queda caracterizada por su tipo  $t_i$  y por su ubicación  $p_i$ . La solución óptima es el conjunto mínimo de estaciones base que satisface simultáneamente las tres condiciones siguientes.

1. Mínima radiación electromagnética sobre la zona  $A$ .
2. Mínimo coste de la instalación o despliegue de red.
3. Máximo grado de cobertura sobre los puntos de interés  $\mathcal{P}'$ , o lo que es equivalente, mínimo porcentaje de no cobertura sobre la zona de estudio  $A$ .

El problema tiene al menos una solución, la óptima global, y otras posibles soluciones óptimas locales. El objetivo consiste en obtener una solución bastante cercana a la óptima con un tiempo de cómputo razonable. Basándonos en los tres criterios anteriores definimos la función de evaluación o función de *fitness* mediante la Ecuación 3.1. Esta ecuación expresa la suma de tres términos donde cada sumando hace referencia a uno de los tres criterios que se desean minimizar:

- $\Delta E_T$  es el incremento total de campo eléctrico en el área  $A$  debido a la instalación completa de la red de telecomunicación.
- $\mathcal{C}_{\mathcal{T}}$  representa el coste total del despliegue obtenido como la suma de los costes unitarios de las estaciones base.
- La variable  $P_{no\ cober}$  es el porcentaje de puntos de  $\mathcal{P}'$  que quedan sin cobertura.

Dado que se tratan de magnitudes bien diferentes, para realizar la adición entre ellas definimos los coeficientes  $k_1$ ,  $k_2$  y  $k_3$ , que multiplicados por la magnitud asociada generan cantidades adimensionales, las cuales pueden ser sumadas entre sí. El coeficiente  $k_1$  tiene unidades de  $(mV/m)^{-1}$ , el coeficiente  $k_2$  se mide en  $(\cdot 10^3 \text{€})^{-1}$  y el coeficiente  $k_3$  es adimensional puesto que multiplica a un porcentaje.

$$\mathcal{F} = k_1 \cdot \Delta E_T + k_2 \cdot \mathcal{C}_{\mathcal{T}} + k_3 \cdot P_{no\ cober} \quad (3.1)$$

A continuación explicamos el modo para calcular cada uno de los términos de la función de evaluación. En este trabajo vamos a considerar coincidentes los conjuntos  $\mathcal{O}$ ,  $\mathcal{P}$  y  $\mathcal{P}'$ . En el B comentamos la siguiente etapa de este trabajo donde los tres conjuntos son disjuntos. Ocurre esta situación cuando, por ejemplo, nuevas instalaciones de estaciones base se restringen a un conjunto de localizaciones independiente de las ubicaciones donde se desea ofrecer el servicio de telecomunicación, que suele ser un conjunto mucho más numeroso. Por otra parte, las medidas experimentales de campo no tienen por qué coincidir con cualquiera de los anteriores conjuntos. Reiteramos que para el trabajo presentado consideramos los tres conjuntos equivalentes a modo de simplificación del problema, sin por ello dejar de tenerlo en cuenta para los trabajos futuros.

Nuestro objetivo es encontrar una solución de despliegue tal que, ofreciendo un servicio aceptable, produzca la mínima radiación electromagnética posible. Conocidas las medidas de campo eléctrico en el conjunto  $\mathcal{O}$ , podemos trasladar el requisito anterior al objetivo de minimizar el incremento total de campo eléctrico sobre  $\mathcal{O}$  debido a la instalación de la nueva red. Tenemos, por tanto, que cuantificar el incremento de campo eléctrico en  $\mathcal{O}$  originado por red compuesta de  $N$  estaciones base.

El campo eléctrico total  $E_{Ti}$  en un punto  $i$  debido al campo inicial junto con el efecto de las  $N$  estaciones base de la nueva red puede calcularse como la suma cuadrática de ambos factores. La Ecuación 3.2 formula la suma cuadrática del campo inicial en el punto  $i$  y el campo en dicho punto creado por las  $N$  estaciones base, siendo  $E_{n,i}$  el campo creado por la estación base  $n$  en el punto  $i$ .

$$E_{Ti} = \sqrt{E_{oi}^2 + \left( \sum_{n=1}^N E_{n,i} \right)^2} \quad (3.2)$$



La Ecuación 3.3 identifica el incremento total de campo eléctrico  $\Delta E_T$  en el área de estudio  $A$ , como el sumatorio de los incrementos parciales de campo eléctrico contabilizados en cada punto del conjunto  $\mathcal{O}$ , donde  $\Delta E_i$  representa el incremento de campo eléctrico para el punto  $i$  debido al conjunto de las  $N$  estaciones base de la red. Dicho incremento se computa como la diferencia entre el campo eléctrico total presente en el punto  $i$  con la red de telecomunicación en servicio ( $E_{Ti}$ ) y el campo eléctrico inicial, es decir, el campo medido en el punto  $i$  antes de la instalación de las nuevas estaciones base ( $E_{oi}$ ).

$$\Delta E_T = \sum_{i=1}^{\mathcal{O}} \Delta E_i = \sum_{i=1}^{\mathcal{O}} (E_{Ti} - E_{oi}) \quad (3.3)$$

La tarea ahora consiste en expresar matemáticamente la magnitud  $E_{n,i}$  de la Ecuación 3.2, que se corresponde con el campo eléctrico en un punto debido a una estación base. La Ecuación 3.4 relaciona la potencia recibida en un punto  $P_R$  con la densidad de potencia  $\phi$  medida en  $W/m^2$  y el área efectiva de la antena receptora  $A_E$ . La Ecuación 3.5 expresa la densidad de potencia  $\phi$  en función del campo eléctrico  $E$ .

$$P_R = \phi A_E \quad (3.4)$$

$$\phi = \frac{E^2}{120\pi} \quad (3.5)$$

La Ecuación 3.6 permite calcular el área efectiva de la antena receptora a partir de la longitud de onda en el espacio libre  $\lambda$  y la ganancia  $G_i$  de la antena con respecto a la antena isotrópica (consideramos  $G_i = 1$ ). La longitud de onda  $\lambda$  es el cociente entre la velocidad de la luz en el vacío ( $c = 3 \times 10^8 m/s$ ) y la frecuencia (Hz),  $\lambda = c/f$ .

$$A_E = \frac{\lambda^2}{4\pi} G_i \quad (3.6)$$

Sustituyendo las Ecuaciones 3.5 y 3.6 en la Ecuación 3.4 obtenemos la Ecuación 3.7, referida al campo eléctrico  $E$  generado por una estación base.

$$P_R = \frac{E_r^2}{120\pi} \frac{\lambda^2}{4\pi} G_i \quad (3.7)$$

Sustituyendo esta expresión en la Ecuación 3.2 se obtiene la Ecuación 3.8, que será la que finalmente sutituiremos en la Ecuación 3.3 para el cálculo del término  $\Delta E_T$  de la función de evaluación. Para ello necesitamos obtener la potencia recibida  $P_{R_{n,i}}$  en cada punto  $i$  de cada una de las  $n$  estaciones base de la red solución.

$$E_{Ti} = \sqrt{E_{oi}^2 + \sum_{n=1}^N \frac{P_{R_{n,i}} 4\pi^2}{\lambda^2 G_i} 120} \quad (3.8)$$

Omitiendo los subíndices  $(n,i)$  de la potencia recibida  $P_{R_{n,i}}$  que identifican la ubicación  $i$  y la estación base  $n$ , el término  $P_R$  lo calculamos a partir de la potencia transmitida por la antena  $P_T$  y las pérdidas de propagación  $L_p$ . La Ecuación 3.9 relaciona estas magnitudes en unidades logarítmicas.

$$P_R(dBW) = P_T(dBW) - L_p(dB) \quad (3.9)$$

Para obtener las pérdidas de propagación utilizamos el modelo *Okumura-Hata* [54] o el modelo *Cost231-Hata* [36] según sea la frecuencia de trabajo de las estaciones base de la red de telefonía. El modelo *Okumura-Hata* es válido para bajas frecuencias,  $150 \leq f \leq 1500 \text{ MHz}$  (GSM que emplea una frecuencia de 900 MHz) mientras que el modelo *Cost231-Hata* es aplicable en frecuencias superiores,  $1500 \leq f \leq 2000 \text{ MHz}$ , GSM 1800 MHz y UMTS 2100 MHz tienen este rango de frecuencias. Ambos modelos estiman las pérdidas de propagación  $L_p$  en función de la distancia  $d(\text{Km})$  desde la estación base hasta un punto receptor, la frecuencia de transmisión  $f(\text{MHz})$ , la altura efectiva de la antena  $h_{BS}$  y la altura del terminal receptor  $h_{Mobile}$ .

La expresión de  $L_p$  para el modelo de *Okumura-Hata* figura en la Ecuación 3.10. En ella, el término  $a(h_{Mobile})$  es un factor de corrección de la altura de la antena medido en metros (m) y definido en la Ecuación 3.11.

$$\begin{aligned} L_p = & 66,95 + 26,16 \cdot \log(f) \\ & -13,82 \cdot \log(h_{BS}) \\ & -a(h_{Mobile}) \\ & +(44,9 - 6,55 \cdot \log(h_{BS})) \cdot \log(d) \end{aligned} \quad (3.10)$$

$$a(h_{Mobile}) = (1,1 \cdot \log(f) - 0,7) \cdot h_{Mobile} - (1,56 \cdot \log(f) - 0,8) \quad (3.11)$$

El modelo *Cost231-Hata* para frecuencias de trabajo superiores a 1500 MHz se recoge en las Ecuaciones 3.12 y 3.13. El factor  $a(h_{Mobile})$  se calcula mediante la Ecuación 3.11. El término  $C_m$  es un factor de corrección que depende del tipo de área en estudio (zonas urbanas o metropolitanas)

$$\begin{aligned} L_p = & 46,3 + 33,9 \cdot \log(f) \\ & -13,82 \cdot \log(h_{BS}) \\ & -a(h_{Mobile}) \\ & +(44,9 - 6,55 \cdot \log(h_{BS})) \cdot \log(d) + C_m \end{aligned} \quad (3.12)$$

$$C_m = \begin{cases} 0 \text{ dB} & \text{ciudades tamaño medio y áreas suburbanas} \\ 3 \text{ dB} & \text{áreas metropolitanas} \end{cases} \quad (3.13)$$

La segunda de las restricciones impuestas a la función de evaluación y enumeradas en la página 50 consiste en encontrar una solución con coste mínimo. El coste del despliegue de red  $C(D)$  se cuantifica de forma simple como la suma de los costes debidos a las  $N$  estaciones base que componen la red solución. En general podemos definir la función de coste como  $C : D \rightarrow \mathbb{R}$ , donde a cada despliegue de red  $D$  le corresponde un coste económico medido en euros (€). La Ecuación 3.14 caracteriza el coste  $C_i(p_i, t_i)$  de una estación base  $i$ , siendo  $p_i$  la ubicación de dicha estación en el área  $A$  ( $p_i \in \mathcal{P}$ ) y  $t_i$  el tipo de estación base, dependiente de la frecuencia de trabajo y la potencia de transmisión. La ubicación  $p_i$  definida por sus coordenadas  $(x_i, y_i)$  es determinante en la función de coste ya que la instalación de equipos en determinadas localizaciones conlleva costes añadidos. En los terrenos privados suele firmarse un acuerdo por el cual el propietario recibe una retribución económica en concepto de utilización para la instalación del equipamiento por parte del operador.

$$C(D) = \sum_{(p_i, t_i) \in D} C_i(p_i, t_i), \quad (3.14)$$

El tercer sumando de la función de evaluación de la Ecuación 3.1 es proporcional al porcentaje de no cobertura de la red sobre el conjunto de puntos de interés  $\mathcal{P}'$ . Dicho porcentaje se determina por el número de puntos de  $\mathcal{P}'$  que quedan fuera del área de cobertura de todas las estaciones base de la red y por lo tanto quedan sin servicio. Una solución  $D$  próxima la óptima tendría un reducido porcentaje de ubicaciones sin servicio definido como  $g(D)$ , o lo que es lo mismo, un elevado porcentaje de cobertura sobre la zona de interés. La Ecuación 3.15 define el grado de no cobertura del despliegue  $D$  como el ratio entre el número total de puntos sin servicio  $nc$  y el número total de puntos de interés  $M$  o cardinal del conjunto  $\mathcal{P}'$ .

$$g(D) = \frac{nc}{|\mathcal{P}'|} = \frac{nc}{M} \quad (3.15)$$

Para determinar si un punto queda sin servicio de red comprobamos que quede fuera de la zona de cobertura de todas las estaciones base de la solución y para ello calculamos el radio de alcance  $d(\text{Km})$  de cada una de ellas. Una localización  $p$  quedará no cubierta por una estación base  $i$  si la distancia entre ambas  $D_{p,i}$  supera el radio de alcance de la estación ( $D_{p,i} > d$ ).

Por otra parte, un punto se encuentra dentro del alcance de una estación base si se cumple la condición descrita por la Ecuación 3.16, donde  $L_p$  son las pérdidas de propagación,  $P_T$  representa la potencia de transmisión de la antena emisora y  $S_r$  es la sensibilidad del receptor. La sensibilidad depende del tipo de servicio, ancho de banda y tecnología empleada, en nuestro caso tomamos  $S_r = -94 \text{ dBm}$ . Para calcular el radio de alcance de una estación base se sustituye el valor máximo para las pérdidas de propagación que cumple la Ecuación 3.16 en la Ecuación 3.10 para el modelo *Okumura-Hata* o en la Ecuación 3.12 para el modelo *Cost231-Hata*, dependiendo de la frecuencia de trabajo la antena, obteniendo de esta forma el radio máximo de alcance de la estación base  $d(\text{Km})$ .

$$L_p(\text{dB}) \leq P_T(\text{dBW}) - S_r(\text{dBm}) + 30 \quad (3.16)$$



## Capítulo 4

# Métodos desarrollados para la resolución del problema MNDP

En este capítulo describimos con detalle cada una de las funciones del algoritmo evolutivo (EA) implementado en este trabajo de tesis para la resolución del problema de optimización MNDP. A continuación incluimos una breve descripción de los otros tres algoritmos implementados: algoritmo *greedy*, algoritmo *Particle Swarm Optimization* (PSO), y algoritmo *Harmony Search* (HS). Hemos utilizado estos algoritmos para comparar sus resultados con los del EA y poder argumentar si este algoritmo es apto o no para la resolución del problema MNDP.

El segundo algoritmo clave de este trabajo, el algoritmo *Coral Reefs Optimization* (CRO) se presenta en el siguiente capítulo, tanto su base teórica como su implementación.

### 4.1 Algoritmo evolutivo desarrollado para el problema MNDP

En esta sección vamos a describir con detalle cada una de las características del algoritmo evolutivo implementado para la resolución del problema MNDP. La codificación empleada para la representación de las soluciones como individuos pertenecientes a una población es una importante característica que puede favorecer o perjudicar notablemente tanto la eficacia como la eficiencia del algoritmo. La codificación utilizada es el primer asunto que abordamos en cada uno de los algoritmos de este capítulo. A continuación exponemos el resto de funciones y operadores característicos del EA: generación de la población inicial con la que se comienza el algoritmo, proceso de selección de individuos, operadores de cruce y mutación, función de evaluación y condiciones de parada del algoritmo.

#### 4.1.1 Codificación empleada

La codificación utilizada está planteada para simplificar en lo posible los operadores de cruce y mutación. En cada generación del algoritmo, estas operaciones se ejecutan un número muy elevado de veces. De modo que cuanto más sencillas resulten, tanto más veloz será el proceso global del EA. Por otro lado, hemos optado por una codificación en la que cada individuo representa unívocamente una solución al problema y viceversa. Es más, en nuestro caso cada individuo representa una única solución y siempre viable. No existen individuos no válidos y por lo tanto no precisamos de ninguna función de reparación de los mismos. Esta característica es adquirida gracias a una codificación que así lo permita junto con una adecuada definición de los conjuntos  $\mathcal{P}$  y  $\mathcal{T}$ . Recordemos del Capítulo 3 que  $\mathcal{P}$  está formado por todas las localizaciones donde es posible instalar las estaciones base, mientras que  $\mathcal{T}$  recoge los distintos tipos de estaciones base

disponibles para la red. Cada individuo se codifica como un conjunto de identificadores numéricos asociados por parejas de la forma:

$$c = \{(p_i|t_i)\}, \quad i = 1, \dots, N,$$

El individuo  $c$  se compone de  $N$  estaciones base. La estación base  $i$  está caracterizada por la pareja  $(p_i|t_i)$ , donde  $p_i \in \mathcal{P}$  es el identificador de su ubicación en la zona de estudio  $A$  y  $t_i \in \mathcal{T}$  es el identificador del tipo de la estación base  $i$ . El tipo de antena está caracterizado por la frecuencia de trabajo, la potencia de transmisión, el coste económico y en algunos casos de la capacidad de conexión de usuarios. En la práctica, la representación de cada individuo figura con las parejas  $(p_i|t_i)$  desglosadas, escribiendo primero todos los identificadores de las ubicaciones  $p_i$  en orden creciente, seguidos de los tipos de las estaciones base  $t_i$  en el mismo orden. Como ejemplo proponemos la codificación para el siguiente individuo formado por 4 estaciones base:

$$[2 \ 18 \ 24 \ 32 \ | \ 3 \ 1 \ 2 \ 1]$$

Este vector de números enteros es la representación de una solución que comprende de una estación base situada en la localización identificada como 2 en el conjunto  $\mathcal{P}$  y cuyo tipo se corresponde con el 3 dentro del conjunto  $\mathcal{T}$ , otra estación ubicada en la localización 18 y de tipo 1, una tercera estación en la localización 24 correspondiente al tipo 2 y una última estación base en el punto número 32 y con tipo 1 dentro del conjunto  $\mathcal{T}$ .

Por último queremos destacar una importante cualidad de la codificación empleada: permite representar individuos de longitud variable, es decir, individuos con diferente número de estaciones base. Para ello escribimos el individuo con sus ubicaciones  $p_i$  siempre ordenadas de menor a mayor. De esta forma evitamos duplicidades en el sentido de que una misma solución se corresponda con más de un individuo.

#### 4.1.2 Proceso de comienzo del algoritmo

El EA comienza generando un número  $N_{ind}$  de individuos aleatorios que forman la generación inicial con la que se dan paso a las operaciones de selección, cruce o recombinación y mutación de la primera generación. Mientras no se diga lo contrario, el valor de  $N_{ind}$  permanece constante durante todo el algoritmo. Cada uno de los  $N_{ind}$  individuos son completamente aleatorios. Esto significa que tienen un número aleatorio de estaciones base situadas al azar en ubicaciones pertenecientes a  $\mathcal{P}$  y con tipos de antenas también aleatorios, dentro del conjunto  $\mathcal{T}$ .

En primer lugar es necesario establecer el rango disponible para el número de estaciones base que componen cada nuevo individuo aleatorio. Dicho rango depende fundamentalmente de las dimensiones del área de estudio  $A$  y del conjunto  $\mathcal{T}$ , puesto que antenas con mayor potencia de transmisión cubren superficies más extensas, y por lo tanto se requiere un número menor de estaciones base para ofrecer servicio en un área  $A$  determinado.

Consideremos el intervalo  $[N_{MinB}, N_{MaxB}]$ , siendo éstos los límites inferior y superior para el rango de número de estaciones base de un individuo. Para sobre-estimar este rango consideremos la situación teórica con mayor número de estaciones base que cubran completamente el área  $A$ . Esta situación teórica se corresponde con la asignación de antenas con la menor potencia de transmisión disponible en el conjunto  $\mathcal{T}$ . Asumiendo que el área de cobertura de una estación base es circular y de valor  $\pi r_{cob}^2$ , definimos  $r_{cob}$  como el radio de cobertura de la antena. Dicho radio depende de la altura a la que se instale la antena, de su potencia de transmisión y de la frecuencia de trabajo, tal y como se explicó en el Capítulo 3 según los modelos *Okumura-Hata* y *Cost231-Hata*. Dado que buscamos una sobre-estimación simplificada de  $[N_{MinB}, N_{MaxB}]$ , consideramos solo la menor altura registrada en el conjunto  $\mathcal{P}$  para el cálculo de  $r_{cob}$  y asumimos

el mismo radio para todas las estaciones base. Definamos  $B_{max}$  como el número de estaciones equivalentes base bajo las condiciones más desfavorables que satisface la restricción de cobertura.  $B_{max}$  es el resultado del ratio entre  $A$  y el área de cobertura  $\pi r_{cob}^2$  de una de ellas tal y como muestra la Ecuación 4.1.

$$B_{max} = \frac{A(m^2)}{\pi r_{cob}^2} \quad (4.1)$$

En la práctica el valor de  $B_{max}$  resulta ser todavía bastante sobre-estimado, indicando un número de estaciones base muy superior al necesario. Por ello resulta conveniente aplicar los factores de reducción de las Ecuaciones 4.2 y 4.3, donde se redondea al entero superior correspondiente puesto que  $N_{MinB}$  y  $N_{MaxB}$  representan números de estaciones base. En el caso de que tras aplicar el factor de reducción resultara un valor de  $N_{MaxB}$  notablemente superior al necesario o bien un valor de  $N_{MinB}$  inferior al requerido experimentalmente, sería responsabilidad del procedimiento de búsqueda local el ajustar el número de estaciones base de sus individuos en el transcurso de la evolución del algoritmo.

$$N_{MaxB} = \lceil 0,4B_{max} \rceil \quad (4.2)$$

$$N_{MinB} = \lceil 0,33N_{MaxB} \rceil \quad (4.3)$$

A continuación exponemos un ejemplo numérico para el cálculo del rango  $[N_{MinB}, N_{MaxB}]$  para una red móvil de 1800 MHz. A partir de la menor potencia de transmisión en el conjunto  $\mathcal{T}$  (6 dBW) y la menor altitud presente en el conjunto  $\mathcal{P}$  (15 m), se obtiene el radio de cobertura teórico mediante el modelo *Cost231-Hata*. El valor numérico en el ejemplo para este parámetro es  $r_{cob} = 457,3673$  m. Con él se calcula  $B_{max}$  según la Ecuación 4.1 y se aplican los factores de reducción de las Ecuaciones 4.2 y 4.3, resultando finalmente el intervalo [9-25] estaciones base para la generación de individuos aleatorios.

- Área  $A = 40,281745$   $Km^2$ ;
- Mínima potencia de transmisión registrada en el conjunto  $\mathcal{T}$ :  $P_T = 6$  dBW
- Frecuencia de trabajo de las estaciones base en  $\mathcal{T}$ : 1800 MHz
- Altura mínima para localizaciones en el conjunto  $\mathcal{P}$ :  $h_{BTS} = 15$  m
- Radio de cobertura calculado mediante la ecuación que describe el modelo de *Cost231-Hata* en el Capítulo 3:

$$\begin{aligned} r_{cob} &= 457,3673 \text{ m} \\ B_{max} &= \lceil \frac{A}{\pi r_{cob}^2} \rceil = 62 \text{ estaciones base} \\ N_{MaxB} &= \lceil 0,4B_{max} \rceil = 25 \text{ estaciones base} \\ N_{MinB} &= \lceil 0,33N_{MaxB} \rceil = 9 \text{ estaciones base} \end{aligned}$$

Una vez que tengamos establecidos el valor numérico para  $N_{ind}$  y el rango  $[N_{MinB}, N_{MaxB}]$ , se generan los  $N_{ind}$  individuos aleatorios, teniendo cada uno de ellos un número aleatorio de estaciones dentro del intervalo  $[N_{MinB}, N_{MaxB}]$ . La localización y tipo de las mismas se selecciona al azar de entre los posibles elementos de los conjuntos  $\mathcal{P}$  y  $\mathcal{T}$ , respectivamente. El Algoritmo 1 recoge todas las acciones que lleva a cabo el proceso de inicialización del algoritmo.

**Algorithm 1** Pseudo-código del proceso de generación de la población inicial en el EA**Require:** Number of individuals  $N_{ind}$  and range  $[N_{MaxB}, N_{MinB}]$ **Ensure:** Feasible random population

- 1: **for** each individual  $i$  of  $N_{ind}$  **do**
- 2:   Generate a random number of BTSs  $N_i \in [N_{MaxB}, N_{MinB}]$
- 3:   **for** each BTS of  $N_i$  **do**
- 4:     Select a random location for the BTS from  $\mathcal{P}$
- 5:     Select a random type for the BTS from  $\mathcal{T}$
- 6:   **end for**
- 7:   Ordering locations in ascending order
- 8: **end for**

Estos individuos distarán mucho del óptimo global. Lo estadísticamente probable es que tengan estaciones base con zonas de cobertura coincidentes y que haya puntos del conjunto  $\mathcal{P}'$  que no sean alcanzados por ninguna estación. Sucesivas generaciones del EA permitirán a los individuos evolucionar, identificando progresivamente soluciones de mejor calidad (valor de *fitness*).

### 4.1.3 Proceso de selección de la población

En el EA distinguimos dos momentos diferentes para efectuar una selección entre los individuos de la población: el proceso de selección de los progenitores antes de la operación de cruce o recombinación y el proceso de selección para formar la población que pasará a la siguiente generación. La selección de padres organiza las parejas que generarán descendencia correspondiente a esa generación. Este tipo de selección la abordamos en la Sección 4.1.4. Los descendientes competirán con la totalidad de la población, incluyendo progenitores, antes de evolucionar a la siguiente generación.

La selección de individuos que sobrevive de una generación a otra se realiza mediante torneo en grupos de dos. Hemos implementado una función de agrupamiento de individuos parametrizable en la que se puede variar tanto el tamaño de los grupos como el número de grupos a obtener, con o sin repetición de individuos <sup>1</sup>. Los resultados expuestos en el Capítulo 6 se obtuvieron realizando esta selección mediante torneos de parejas. El torneo por parejas para la selección de la población de la generación posterior funciona de la siguiente manera:

Denominemos  $W_i$  a población que alcanzó la generación  $i$ ; asimismo llamemos  $D_i$  a los descendientes de la generación  $i$ . La población que sobrevive una generación más ( $W_{i+1}$ ) se construye con los vencedores del torneo entre todos ( $W_i + D_i$ ). Se forman tantas parejas de individuos sin repetición como sea posible ( $N_g = \frac{W_i + D_i}{2}$ ). De cada pareja, el ganador será el individuo con el mejor valor de *fitness*. La población que sobrevive  $W_{i+1}$  se forma a partir de los ganadores del torneo. Si no se ha alcanzado el tamaño  $N_{ind}$  de la población, ésta se completa con los mejores individuos de la población participante en el torneo ( $W_i + D_i$ ) que no estuvieran ya incluidos en  $W_{i+1}$ , es decir, se seleccionan de forma elitista entre los participantes.

<sup>1</sup>La función de agrupamiento de individuos se emplea tanto para la formación de grupos para el torneo como para la formación de parejas de progenitores. Se ha implementado con parámetros variables que permiten la formación de  $N_g$  grupos de determinado tamaño (2 elementos si se trata de parejas) con o sin repetición de elementos entre grupos y dentro de cada grupo (en el caso de parejas para la selección carece de sentido permitir la repetición de elementos dentro de un mismo grupo puesto que tendríamos un único individuo que compite con él mismo).



#### 4.1.4 Proceso de selección de progenitores

Los dos métodos implementados para esta operación son: formación de parejas con individuos equiprobables y el método de la ruleta. El método básico de formación de parejas de progenitores en la generación  $i$  consiste en ir seleccionando aleatoriamente de  $W_i$  el número de parejas de individuos  $N_g$  que esté establecido. Utilizamos aquí la misma función de agrupamiento parametrizable mencionada para la selección de grupos de torneo. En este caso se puede permitir que un mismo individuo participe en más de una pareja de modo que sea progenitor de más de un descendiente con la misma o distintas parejas. Esto significa permitir repetición entre grupos. Sin embargo carece de sentido permitir repetición de individuos dentro de una misma pareja según la definición de los operadores de recombinación en la Sección 4.1.5. Asumimos equiprobabilidad en la formación de parejas, en el sentido de que todos los individuos tienen la misma probabilidad de ser seleccionados como progenitor de una pareja.

La selección mediante el método de la ruleta consiste en asignar a cada individuo una probabilidad de selección proporcional a la posición que ocupe su valor de *fitness* en el rango de *fitness* de la población. De esta forma, los individuos con mejor valor de la función de evaluación tienen mayores posibilidades de ser elegidos como progenitores que los individuos con peor valor de la función objetivo. Acompañamos la explicación con un ejemplo numérico recogido en la Tabla 4.1 y en la Figura 4.1. Este ejemplo tiene cierta semejanza con el expuesto en el Capítulo 2 pero con la importante diferencia de aquí el objetivo es minimizar la función de evaluación y no maximizarla como en aquel caso. Supongamos una población compuesta por seis individuos, tal que  $N_{ind} = 6$ . La primera columna de la Tabla 4.1 indica el identificador de cada uno de ellos. En la segunda columna se reflejan los valores de *fitness*  $\mathcal{F}$  de cada individuo  $i$ . La siguiente columna calcula el inverso de  $\mathcal{F}$ . La última fila de la tabla muestra la suma total de los inversos de  $\mathcal{F}$ . El valor de esta suma (0,2 en el ejemplo) se hará corresponder con el máximo de una probabilidad, es decir, la unidad. La cuarta columna de la tabla calcula la probabilidad de selección  $p_i$  del individuo  $i$  de acuerdo a dicha proporción mediante la Ecuación 4.4. El nombre de “ruleta” para este método hace referencia al reparto de probabilidades como si de una tarta o pastel se tratara. La Figura 4.1 representa la distribución de probabilidades por sectores circulares según la última columna de la Tabla 4.1, que expresa el ángulo del sector circular que le corresponde al individuo  $i$ , siempre proporcional a  $p_i$ .

Es necesario destacar que para aplicar este método de selección de individuos por ruleta, el valor de *fitness* no puede ser nulo en ningún individuo, puesto que forzaría a que su valor inverso  $1/\mathcal{F}_i$  fuera infinito y el método no resultaría realizable.

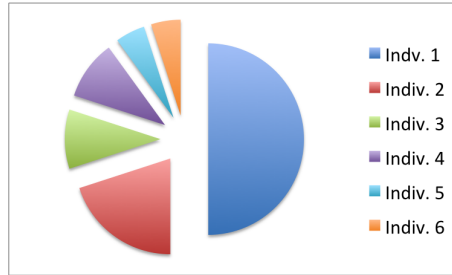
$$p_i = \begin{cases} \frac{1/\mathcal{F}_i}{\sum_{i=1}^{N_{ind}} \frac{1}{\mathcal{F}_i}}; & i = 1 \\ p_{i-1} + \frac{1/\mathcal{F}_i}{\sum_{i=1}^{N_{ind}} \frac{1}{\mathcal{F}_i}}; & 1 < i \leq N_{ind} \end{cases} \quad (4.4)$$

#### 4.1.5 Operador de cruce o recombinación

La operación de cruce sirve para generar nuevos individuos a partir de los ya existentes en la población. Se pretende además que los de nueva creación superen a sus progenitores en calidad, es decir, con mejor valor de la función de evaluación. Todos juntos competirán por alcanzar la siguiente generación, venciendo probablemente los de mejor valor de *fitness*. De esta manera, el algoritmo evolucionará con poblaciones cada vez más próximas al individuo óptimo. Si los individuos fueran creados simplemente de forma aleatoria en vez de basarse en el contenido de

**Tabla 4.1:** Ejemplo de distribución de probabilidades de selección de progenitores por ruleta para una población de seis individuos a partir de sus valores de *fitness*.

Individuo $i$	$\mathcal{F}_i$	$1/\mathcal{F}_i$	Probabilidad $p_i$	Ángulo sector circular (°)
1	10	0,1	0,5	180
2	25	0,04	0,2	72
3	50	0,02	0,1	36
4	50	0,02	0,1	36
5	100	0,01	0,05	18
6	100	0,01	0,05	18
$\sum_{i=1}^6$		0,2	1	360

**Figura 4.1:** Ejemplo de distribución de probabilidades de selección de individuos por el método de la ruleta para la población de la Tabla 4.1.

los progenitores, el algoritmo no sería evolutivo, sino que realizaría una búsqueda de soluciones completamente aleatoria.

A partir de dos individuos progenitores se construyen uno o dos individuos nuevos que contienen una parte de información de cada progenitor. Estadísticamente, en unos casos los individuos descendientes representan soluciones mejor adaptadas (mejor valor de *fitness*), mientras que en otros casos representan soluciones peores que las de sus progenitores. La función de búsqueda local se encarga de mejorar algunos individuos descendientes mientras que el proceso de selección otorga mayor probabilidad de supervivencia a los individuos con mejor valor de *fitness*.

En este trabajo hemos desarrollado dos métodos específicos para el cruce de individuos en el problema MNDP, los cuales hemos denominado TBC (*Template-Based Crossover*) y DMC (*Direct Merge Crossover*). En ambos casos obtenemos un solo individuo descendiente en cada operación de recombinación. El operador TBC está basado en el uso de una máscara binaria que determina la procedencia de cada gen del descendiente, es decir, qué progenitor transfiere el contenido del gen. El operador DMC consiste en generar un nuevo individuo agrupando la totalidad de información de ambos progenitores, por lo que el descendiente recoge todas las estaciones base que figuran en ambos progenitores. Después, el proceso de búsqueda local permite eliminar estaciones base que no mejoran la calidad de la solución, como por ejemplo, estaciones base que solapan sus zonas de cobertura con otras. A continuación describimos cada uno de los dos operadores de cruce planteados.

#### 4.1.5.1 Operador de cruce TBC

Como ya hemos mencionado, el operador de cruce TBC utiliza una máscara binaria para indicar de qué progenitor debe copiar cada gen en el nuevo individuo. En el problema de optimización MNDP los individuos son de longitud variable, dado que cada solución está compuesta de un número específico de estaciones base. La longitud del individuo descendiente coincide con la del progenitor con mayor número de estaciones base. Denominemos a este individuo progenitor #1. Por otro lado, la longitud de la máscara se hace coincidir con la longitud del progenitor con menor número de estaciones. Llamemos progenitor #2 a este individuo. La interpretación de los bits de la máscara es coherente con la codificación de los individuos. Recordamos que una solución con  $N$  estaciones base se codifica como un conjunto de  $N$  pares  $(p_i|t_i)$  disociados de la forma  $(p_1\ p_2\ \dots\ p_N\ t_1\ t_2\ \dots\ t_N)$ . La mitad izquierda de la máscara hace referencia a localizaciones de estaciones base mientras que la mitad derecha a tipología de éstas. Los “1” de la máscara binaria determina qué genes o estaciones base del progenitor #2 son utilizados para la construcción del nuevo individuo.

El proceso de cruce TBC comienza con un individuo réplica del progenitor #1, asegurando por tanto la misma longitud para el descendiente. Debido a la diferencia de longitud entre progenitores, la máscara tiene cierto rango de desplazamiento entre los genes del progenitor #1. La amplitud de dicho rango queda determinada por la diferencia de longitudes de los progenitores. Es necesario fijar un punto de inicio desde donde comenzar la operación de sustitución de genes. Dicho punto de inicio se aplica de forma idéntica a las dos partes del individuo: localizaciones y tipos de estaciones base. La Figura 4.2 facilita la comprensión del proceso mediante un ejemplo. Partiendo de un progenitor #1 compuesto por 6 estaciones base situadas en las ubicaciones 1, 10, 12, 24, 33 y 44, respectivamente. El progenitor #2 está formado por 4 estaciones base en las localizaciones 2, 18, 24 y 32. La máscara binaria aleatoria dispone de 4+4 bits, los primeros se operarán sobre las localizaciones de las estaciones y los siguientes 4 sobre la tipología de las mismas. El punto de comienzo en el progenitor #1 para aplicar la máscara pertenece al rango [1-3], de forma que todos los genes del progenitor #2 tengan siempre un gen en el primero sobre el que operar la máscara. Si el punto de inicio estuviera fuera de este intervalo, por ejemplo en la posición 4, el último gen del progenitor #2 no tendría gen en el progenitor #1 sobre el que aplicar la máscara, ni en la parte  $p_i$  de localizaciones ni en la parte  $t_i$  de tipos. En el ejemplo se ha seleccionado aleatoriamente el punto de inicio 2, luego el progenitor #2 opera sobre los genes del progenitor #1 en posiciones 2, 3, 4 y 5 tanto en localizaciones como en tipos. Finalmente, el individuo descendiente sustituye sus valores numéricos que se corresponden con los tres “1” de la máscara por los valores del progenitor #2, intercambiando la ubicación 10 por la 2, la 33 por la 32 y el tipo de la tercera estación base pasa de ser 1 a ser 3. Como puede observarse la máscara puede alterar solo la localización de una estación base, solo el tipo de ésta o bien ambos parámetros simultáneamente.

El Algoritmo 2 recoge el pseudo-código para el operador de cruce TBC. Si se desea construir un segundo descendiente a partir de los mismos progenitores, basta realizar la operación con el complementario de la máscara binaria, es decir, considerar un “1” donde había un “0” y viceversa. El pseudo-código finaliza con la ordenación de genes en orden ascendente de localizaciones y una llamada al proceso de búsqueda local sobre el nuevo individuo. Mantener ordenadas las localizaciones asegura que a cada solución le corresponda una única codificación de individuo. El proceso de búsqueda local permite mejorar la calidad de los individuos evitando por ejemplo la inclusión de estaciones base innecesarias. Esta operación puede ser opcional o bien realizarla con cierta probabilidad.

Initial Couple:	1 10 12 24 33 44   3 1 1 3 2 2
	2 18 24 32   1 3 2 1
Template:	1 0 0 1   0 1 0 0
Couple with template:	1 10 12 24 33 44   3 1 1 3 2 2
	1 0 0 1   0 1 0 0
	2 18 24 32   1 3 2 1
Final individual:	1 2 12 24 32 44   3 1 3 3 2 2

**Figura 4.2:** Ejemplo de cruce de tipo TBC de dos individuos en el EA.

---

**Algorithm 2** Pseudo-código del operador de cruce TBC en el EA

---

**Require:** A feasible individuals couple:  $individual_A = (p_1 \ p_2 \dots p_A \ t_1 \ t_2 \dots t_A)$  and  $individual_B = (p_1 \ p_2 \dots p_B \ t_1 \ t_2 \dots t_B)$

**Ensure:** A single feasible offspring

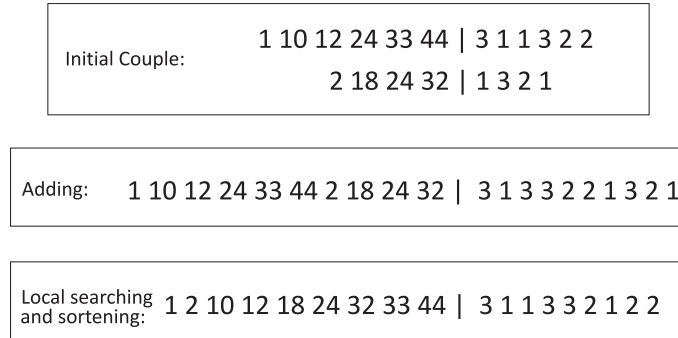
- 1: **if**  $A > B$  **then**
  - 2:    $l_1 = A$
  - 3:    $l_2 = B$
  - 4:    $offspring = (p_1 \ p_2 \dots p_A \ t_1 \ t_2 \dots t_A)$
  - 5: **else**
  - 6:    $l_1 = B$
  - 7:    $l_2 = A$
  - 8:    $offspring = (p_1 \ p_2 \dots p_B \ t_1 \ t_2 \dots t_B)$
  - 9: **end if**
  - 10: Random binary mask with  $l_2$  bits
  - 11: Select crossover point  $p \in \{1, \dots, l_1 - l_2 + 1\}$
  - 12: **for**  $i = p$  to  $(p + l_2 - 1)$  **do**
  - 13:    $Offspring(i) = mask(i)individual_A(i) + (1 - mask)(i)individual_B(i)$
  - 14: **end for**
  - 15: Ordering identifiers of BTSs locations for new offspring
  - 16: Local searching
- 

#### 4.1.5.2 Operador de cruce DMC

El operador de cruce DMC es más sencillo que el operador de cruce TMC, además de ofrecer mejores resultados. Este operador *sobredimensiona* el individuo descendiente incorporando numerosas estaciones base. A continuación delega en el proceso de búsqueda local la decisión de qué estaciones base son las más convenientes para una buena solución a partir exclusivamente de la información contenida en los dos individuos progenitores.

El descendiente recoge todas y cada una de las estaciones base de ambos progenitores, copiando tanto sus localizaciones como los tipos de las mismas. Lo habitual es que varias de estas estaciones resulten innecesarias en la nueva solución, y por ello es importante ejecutar el procedimiento de búsqueda local tras este tipo de cruce. La Figura 4.3 muestra un ejemplo de cruce DMC con los mismos individuos progenitores que en el ejemplo anterior con el operador TBC. Como puede observarse, el descendiente está formado inicialmente por las 10 estaciones

base presentes en los progenitores. En ocasiones puede suceder que ambos progenitores utilicen una misma localización independientemente del tipo de estación asignado. Esto ocurre en el ejemplo con la localización número 24, en el progenitor #1 tiene asignada el tipo 3 mientras que el progenitor #2 figura con el tipo 2. Una vez procesada la búsqueda local y ordenados los genes por orden ascendiente de localización, la solución final registra 9 estaciones base.



**Figura 4.3:** Ejemplo de cruce de tipo DMC de dos individuos en el EA.

El Algoritmo 3 presenta el pseudo-código para el operador de cruce DMC, donde el descendiente se construye concatenando los genes de ambos progenitores. Es importante aplicar la búsqueda local para identificar y retirar genes que no aportan calidad al individuo. Obviamente la búsqueda local añade carga computacional al algoritmo. Resulta muy recomendable controlar el porcentaje de cruces de tipo DMC para alcanzar un compromiso entre eficiencia y eficacia en el EA. En la Sección 6 especificamos el porcentaje de cruce de cada tipo empleado en las simulaciones realizadas.

---

**Algorithm 3** Pseudo-código del operador de cruce DMC en el EA

---

**Require:** A feasible individuals couple

**Ensure:** A single feasible offspring

- 1: offspring = concatenate individuals couple
  - 2: Ordering identifiers of BTSs locations for new offspring
  - 3: Local searching
- 

#### 4.1.6 Operador de mutación

La mutación de un individuo realiza cambios significativos en su contenido. Este operador se aplica en un número muy reducido de ocasiones en el EA, pero permite contribuir a la diversidad de la población y la exploración de nuevas zonas en el espacio de búsqueda. Hemos utilizado el método clásico de mutación mediante máscara binaria. La máscara binaria tiene carácter aleatorio y la misma longitud que el individuo sobre el que se aplica. Los “1” de la máscara indican los genes que se modifican en el individuo con nuevos valores aleatorios, tanto para las localizaciones como para los tipos de las estaciones base. Los “0” de la máscara mantienen invariables los genes del individuo asociados.

La Figura 4.4 muestra un ejemplo de mutación para el individuo resultante del cruce DMC de la Figura 4.3 formado por 9 estaciones base. La máscara binaria aleatoria tiene por tanto 9+9 bits, e indica con sus “1” las posiciones de los genes del individuo que deben ser alteradas. Estas corresponden a las localizaciones de las estaciones base con los identificadores 2, 10, 24 y

33, junto con los tipos de las estaciones base ubicadas en las genes 1, 5, 6 y 8 del individuo. Una estación base puede verse afectada por la mutación solo en su localización, solo en su tipo, en ambos aspectos o bien en ninguno de ellos.

Initial individual:	1 2 10 12 18 24 32 33 44   3 1 1 3 3 2 1 2 2
template:	1 2 10 12 18 24 32 33 44   3 1 1 3 3 2 1 2 2 0 1 1 0 0 1 0 1 0   1 0 0 1 1 0 1 0
mutation:	1 76 19 12 18 52 32 14 44   2 1 1 3 1 3 1 1 2
sortering:	1 12 14 18 19 32 44 52 76   2 1 1 1 1 3 2 3 1

**Figura 4.4:** Ejemplo de mutación de un individuo para el EA.

El pseudo-código para este operador se muestra en el Algoritmo 4. El proceso de mutación finaliza con la ordenación de los genes del individuo resultante, siempre en orden creciente de localizaciones, y la búsqueda local cuando es preciso, con el objeto de mejorar el valor de *fitness* del individuo. La longitud del individuo final no siempre coincide con la del individuo previo a la mutación, ya que la búsqueda local puede eliminar alguna estación base o desplazada a otra ubicación coincidente o muy próxima a otra estación ya existente.

---

**Algorithm 4** Pseudo-código para el operador de mutación en EA

---

**Require:** An individual of length  $L$

**Ensure:** Feasible mutated individual

- 1: Generate random binary mask with length  $L$
  - 2: In locations part  $p_i$ , change locations to other random ones where mask has bit '1'
  - 3: In typing part  $t_i$ , change types to other random ones where mask has bit '1'
  - 4: Ordering stations
  - 5: Local searching
- 

#### 4.1.7 Proceso de búsqueda local

Algunas codificaciones permiten la existencia de individuos que no representan soluciones viables. En estos casos es obligatorio realizar la reparación del genotipo. La codificación que empleamos en nuestro trabajo hace corresponder a cada solución con un único individuo y viceversa. De este modo no se requiere reparar ningún individuo.

Sin embargo, a partir de una población dada, es posible mejorar la calidad de algunos de sus integrantes modificando estratégicamente el contenido de uno o varios genes que constituyen los individuos. Esta operación consiste en analizar la zona vecina de una solución en el espacio de búsqueda y comprobar si existe dentro de ella alguna otra solución con mejor valor de *fitness*. El proceso toma el nombre de búsqueda local, puesto que busca mejores soluciones en el entorno local de una solución determinada. La búsqueda local resulta computacionalmente costosa puesto que tiene que evaluar un conjunto de individuos en el entorno local de otro. Esta carga computacional hace que no deba aplicarse sistemáticamente en toda la población y para cada

generación. Además un uso excesivo de este recurso podría causar la convergencia prematura del algoritmo en algún óptimo local.

En el problema MNDP, una solución concreta puede ser mejorada estudiando si alguna de las estaciones base que la componen no contribuye lo suficiente en la función de evaluación como para justificar su presencia. Es el caso, por ejemplo, de estaciones base que tuvieran la zona de cobertura completamente incluida en el área de cobertura de alguna otra estación base en la solución. Entonces la primera estación contribuye negativamente al valor de *fitness* del individuo: aumentando tanto la radiación electromagnética como el coste económico de la red, sin incrementar el porcentaje de cobertura en la zona de estudio<sup>1</sup>. Retirando dicha estación base, el valor de *fitness* mejorará al reducir tanto la radiación como el coste, con el mismo porcentaje de cobertura. Este es un ejemplo sencillo, también surgen casos menos evidentes de individuos que tras estudiarlos concluimos que compensa retirar o desplazar alguna estación base para acercar dicho individuo al óptimo.

La operación de búsqueda local se ocupa de determinar qué estaciones son prescindibles en el individuo y decidir qué hacer con ellas. Las dos acciones básicas a realizar con una estación prescindible son: eliminarla definitivamente del individuo o desplazarla a otra localización donde resulte más eficiente su uso en la red, en cuanto al valor de *fitness* se refiere. Una estación base se considera prescindible si cumple al menos una de las dos condiciones siguientes:

- Todas las localizaciones a las que da servicio una estación base están a su vez cubiertas por otras estaciones base.
- La estación base ofrece servicio solo a un escaso porcentaje de localizaciones, por debajo de un umbral establecido  $\beta\%$ . Generalmente se toman valores de  $\beta$  inferiores al 1%.

La Figura 4.5 es la representación gráfica de un individuo sobre el que se identifican tres ejemplos de estaciones prescindibles. Estos ejemplos se encuentran señalados en la figura con las etiquetas: “Caso 1”, “Caso 2” y “Caso 3”. El área de estudio  $A$  se corresponde a la ciudad de Alcalá de Henares, de la que se dispone un conjunto  $\mathcal{P}$  compuesto por las 509 localizaciones procedentes del proyecto de investigación presentado en el Capítulo 1. Las localizaciones están indicadas en la figura con puntos de color azul, mientras que las estaciones base están representadas por circunferencias que enmarcan sus zonas teóricas de cobertura. El centro de cada circunferencia corresponde a la ubicación de su estación base. Con el objeto de destacar las estaciones prescindibles, hemos empleado el color rojo para éstas. Las circunferencias de color verde representan a las estaciones base que provocan que otras resulten prescindibles. Por último, las circunferencias que no destacan en la operación de búsqueda local aparecen en color negro.

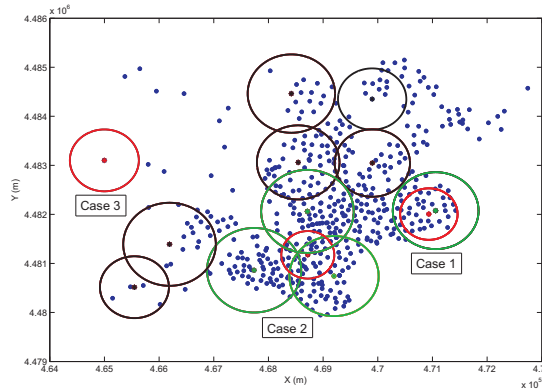
La primera condición definida arriba para que una estación base resulte prescindible en una solución consiste en que todas las localizaciones a las que ofrece servicio estén a su vez cubiertas por una o más estaciones de la misma solución. Esta condición se cumple tanto en el “Caso 1” como en el “Caso 2” de la Figura 4.5. En el primero la circunferencia de color rojo se encuentra completamente inscrita en la circunferencia de color verde, luego todas las localizaciones alcanzadas por la estación base de la circunferencia roja están también incluidas en la zona de cobertura de la estación base representada por la circunferencia de trazo verde. El resultado es que la primera estación base es prescindible.

El “Caso 2” de la Figura 4.5 muestra nuevamente una estación base prescindible porque todas las localizaciones dentro de su zona de influencia, representada por la circunferencia en

<sup>1</sup>Describimos la función de evaluación en la Sección 4.1.8. Se define como la suma de la influencia de tres parámetros: el incremento de campo eléctrico total  $\Delta E_T$  en la zona de estudio  $A$ , el coste total  $\mathcal{C}_{\mathcal{T}}$  del despliegue de las estaciones base de la red y el porcentaje de localizaciones en  $A$  que quedan sin servicio  $P_{no\ cober}$ . La expresión matemática para la función de evaluación o *fitness* es:  $\mathcal{F} = k_1 \cdot \Delta E_T + k_2 \cdot \mathcal{C}_{\mathcal{T}} + k_3 \cdot P_{no\ cober}(\%)$ .

color rojo, se encuentran dentro de al menos una de las zonas de cobertura de las otras tres estaciones, circunferencias de trazo verde.

Por último, el “Caso 3” de la Figura 4.5 es un ejemplo para la segunda de las condiciones que califica a una estación base como prescindible. Definiendo un umbral mínimo  $\beta$  de localizaciones a cubrir para que una estación base no resulte prescindible. En este trabajo hemos utilizado  $\beta = 0,5\%$ . Cubrir menos del 0,5% de 509 localizaciones significa cubrir menos de tres de ellas. En el ejemplo de la figura la estación base cubre una única localización, que coincide con la ubicación de la propia antena y por lo tanto queda calificada como prescindible.

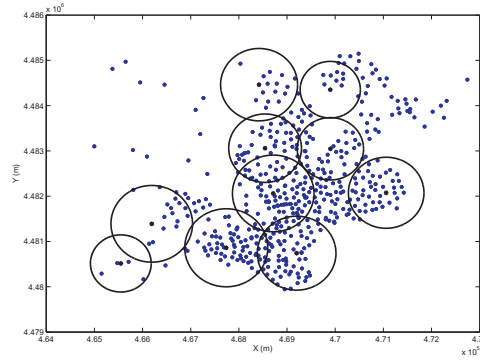


**Figura 4.5:** Representación gráfica de un individuo del problema MNDP sobre el que se aplica el proceso de búsqueda local, identificando tres estaciones base prescindibles destacadas en color rojo.

La Figura 4.6 muestra el resultado final del proceso de búsqueda local en el individuo de la Figura 4.5 cuando se eliminan todas las estaciones base prescindibles. Sin atender a la expresión matemática de la función de evaluación, es fácil argumentar que la solución obtenida es mejor que la inicial. Al retirar tres estaciones base de la red, el coste económico es obviamente inferior. Con tres equipos menos, el incremento total de campo eléctrico en la zona debe ser menor también. El porcentaje de no cobertura no varía significativamente puesto que se deja de ofrecer servicio solo a la localización correspondiente al “Caso 3”. Esto supone un incremento de porcentaje de no cobertura solo del  $1/509=0,196\%$ . Por lo tanto el individuo transformado por el proceso de búsqueda local es de mejor calidad que el individuo del que se parte. Para simplificar el ejemplo expuesto hemos optado por la eliminación de todas las estaciones base prescindibles. Si hubiéramos realizado desplazamiento de algunas de ellas el resultado hubiese sido lógicamente distinto.

Durante el proceso de búsqueda local, la identificación de una estación base como prescindible o no depende del orden en el que se revisen todas las estaciones base del individuo. Si se repite la búsqueda local sobre un mismo individuo siempre en el mismo orden, el resultado será idéntico en todas las ocasiones. Sin embargo, si se establece un orden aleatorio y diferente cada vez, es muy probable obtener resultados diversos. La Figura 4.7 se expone un ejemplo de cómo la operación de búsqueda local sobre un individuo puede tener más de un resultado. La Figura 4.7(a) es la representación gráfica de una solución compuesta por seis estaciones base en la misma zona de estudio *A* que figuras anteriores. En ella aparecen dos estaciones base destacadas en color rojo porque una u otra resultan prescindibles, pero no las dos simultáneamente. Para simplificar el ejemplo, hemos eliminado deliberadamente las estaciones base que no influyen en esta decisión. En este ejemplo no atendemos al valor de la función de evaluación del individuo, sino a identificar





**Figura 4.6:** Representación gráfica del individuo de la Figura 4.5 tras retirar de la solución las tres estaciones base identificadas como prescindibles.

de qué estaciones base podemos prescindir. Dependiendo del orden en el que se revisen estas seis estaciones, la respuesta será la expuesta en la Figura 4.7(b) o bien en la Figura 4.7(c), siempre y cuando la búsqueda local en encargue de retirar de la solución las estaciones prescindibles. En caso de desplazamiento de dichas estaciones a nuevas ubicaciones, el resultado del proceso sería distinto a lo mostrado en las Figuras 4.7(b) y 4.7(c). Lo óptimo sería recoger el mejor resultado de todos los posibles. Sin embargo esto podría provocar la convergencia prematura del algoritmo, además de un muy elevado incremento de la carga computacional, al tener que analizar y evaluar todas las posibles opciones de la búsqueda local sobre un mismo individuo. Por esta razón hemos implementado la operación de búsqueda local en el EA para que considere único un resultado aleatorio en cada ocasión.

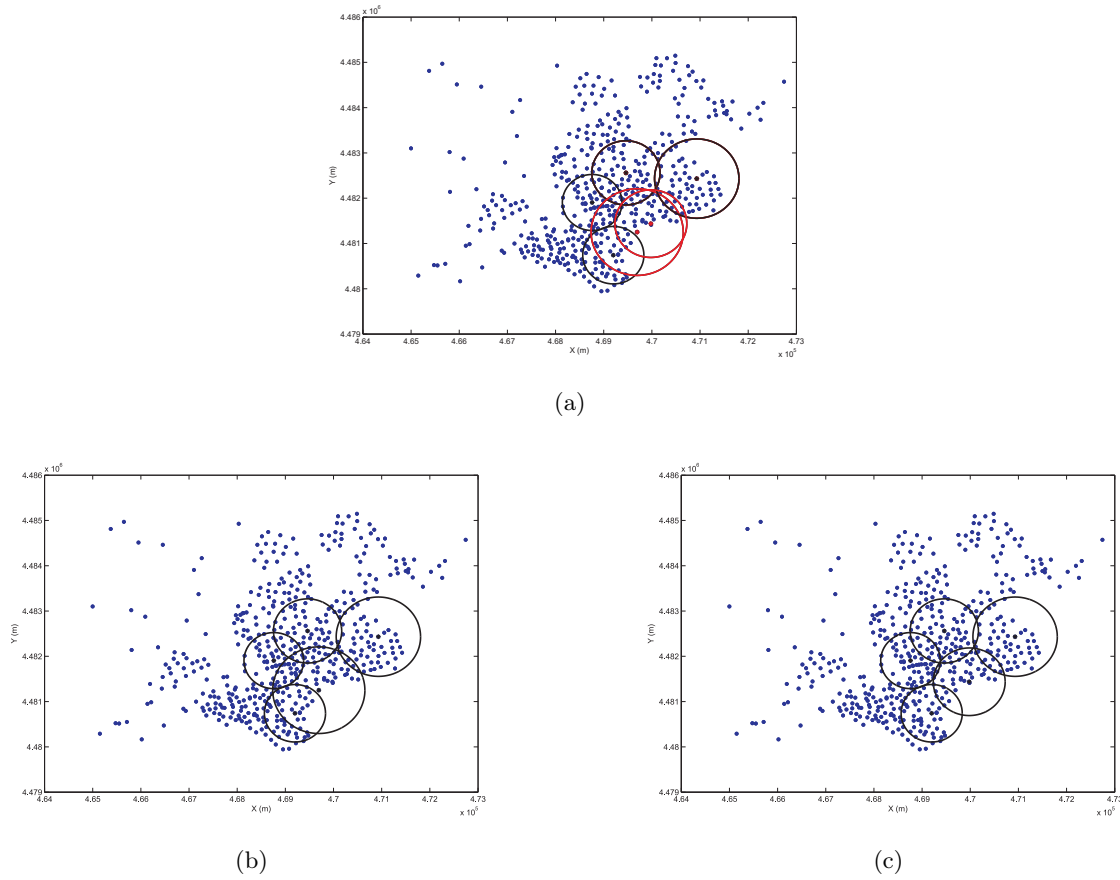
El Algoritmo 20 detalla el pseudo-código de la función de búsqueda local en un individuo eliminando todas las estaciones prescindibles (*expendable stations*), es decir, el pseudo-código aplicado en los ejemplos de las Figuras 4.5 y 4.7. El individuo que se obtiene siempre es igual o mejor que el individuo sobre el que se opera porque en el peor de los casos, ninguna estación base de la solución es prescindible. En caso de eliminar alguna prescindible, el valor de la función de evaluación siempre mejora.

El Algoritmo 20 utiliza la matriz de cobertura ( $N \times M$  *coverage matrix*) del individuo. Está definida como una matriz binaria de  $N$  filas y  $M$  columnas.  $N$  es el número de estaciones base de la solución y  $M$  es el cardinal del conjunto  $\mathcal{P}'$  o número de localizaciones a las ofrecer el servicio de telecomunicación ( $\#\mathcal{P}' = M$ )<sup>1</sup>. Si el elemento  $x_{i,j}$  de la matriz de cobertura tiene el valor unidad significa que la estación base  $i$  cubre la localización  $j$ . El elemento  $x_{i,j}$  relaciona la estación base número  $i$  en el individuo solución, ubicada en las coordenadas  $(x,y)$ , con el punto  $j$  de  $\mathcal{P}'$ , definido por sus coordenadas  $(x'_j, y'_j)$ . Tal y como se indica en la Ecuación 4.5, si la cobertura de la estación base  $i$  alcanza la localización  $j$ , entonces el valor del elemento  $x_{j,j}$  de la matriz es la unidad, en caso contrario su valor es nulo.

$$x_{i,j} = \begin{cases} 1; & \text{si estación } i \text{ ofrece cobertura a la localización } j \\ 0; & \text{si estación } i \text{ no alcanza la localización } j \end{cases} \quad (4.5)$$

La condición para que la localización  $j$  se encuentre dentro de la zona de cobertura de la estación base  $i$  es que la distancia entre los puntos  $(x,y)$  y  $(x'_j, y'_j)$  sea menor que el radio de

<sup>1</sup>Recordamos que en este trabajo consideramos que los conjuntos  $\mathcal{P}$ ,  $\mathcal{P}'$  y  $\mathcal{O}$  son coincidentes, luego  $M$  coincide en este caso con el número total de localizaciones de la zona de estudio  $A$ .



**Figura 4.7:** Ejemplo de cómo la búsqueda local sobre un mismo individuo ofrece más de un resultado. Dependiendo del orden en el que se revisan las estaciones base de la solución al problema MNDP representada en la Figura 4.7(a), una de las dos estaciones base destacadas en color rojo resulta prescindible, pero no las dos a la vez. Las Figuras 4.7(b) y 4.7(c) muestran la solución final tras eliminar una u otra estación.

cobertura teórico de dicha estación. La Figura 4.8 ofrece la representación gráfica para un ejemplo de matriz de cobertura de un individuo solución compuesto por 7 estaciones base sobre un conjunto de 21 localizaciones en  $A$ . Puede observarse que una localización  $j$  se queda sin servicio si la columna  $j$  de la matriz no contiene ningún “1”. En la Figura 4.8 aparecen cuatro localizaciones fuera de servicio, identificadas con los números del 16, 17, 18 y 19.

Por último, el Algoritmo 6 detalla el pseudo-código del proceso de búsqueda local cuando las estaciones base prescindibles son bien eliminadas, o bien desplazadas a nuevas localizaciones. Para catalogar una estación base como prescindible, se comprueba si el número de localizaciones a las que ofrece cobertura de forma exclusiva (ninguna otra estación alcanza la localización) no supera el umbral mínimo definido como  $\beta\%$ . Esta acción queda reflejada en las líneas 9 y 10 del algoritmo.

El objeto de este procedimiento es dotar al EA de mayor diversidad en la población, evitando además que la dimensión de los individuos se reduzca progresivamente según progresa el algoritmo si solo se aplica el Algoritmo 20. En la Sección 6 se especifica la probabilidad asignada a las acciones de eliminar y desplazar las estaciones base prescindibles de un individuo en las simulaciones realizadas para los resultados que se presentan allí.

---

**Algorithm 5** Pseudo-código del procedimiento de búsqueda local en un individuo con eliminación de estaciones base prescindibles

---

**Require:** An individual

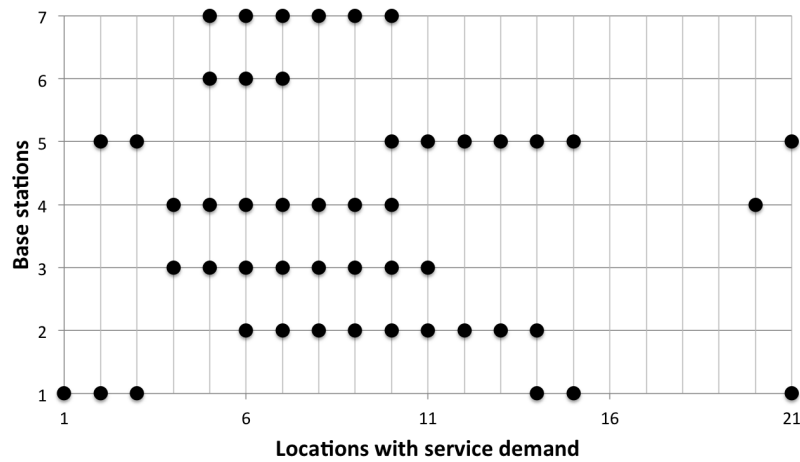
**Ensure:** A similar feasible individual with same or better fitness value

```

1: Generate the  $N \times M$  coverage matrix
2: for each station of the individual do
3:   if less coverage than  $\beta\%$  then
4:     Remove station
5:   end if
6: end for
7: Update the  $N \times M$  coverage matrix
8: Consider all stations are expendable
9: for each  $i$  station in random order do
10:  if any locations in  $A$  is fulfilled by more than one station then
11:    Break this loop
12:  end if
13:  if there is at least one location covered by  $i$  station which is fulfilled by only  $i$  station then
14:    Check  $i$  station as non expendable
15:  end if
16: end for
17: Remove all expendable stations
18: Update the  $N \times M$  coverage matrix

```

---



**Figura 4.8:** Representación de la matriz de cobertura  $N \times M$  de un individuo como posible solución para una zona de estudio  $A$  definida por 21 localizaciones como objetivo de demanda ( $\#\mathcal{P}' = M = 21$ ). La solución consta de 7 estaciones base ( $N=7$ ). La matriz simboliza con puntos negros aquellos elementos  $(i,j)$  donde la localización  $j$  se encuentra dentro de la zona de cobertura de la estación base  $i$ .

#### 4.1.8 Función de evaluación

Como se explicó en el Capítulo 3, la solución al problema de optimización MNDP consiste en determinar un despliegue de red  $D = \{(p_i, t_i)\}, i \in \{1, N\}$  compuesto por  $N$  estaciones base distribuidas en diferentes ubicaciones del área de estudio  $A$  tal que  $D$  se acerque a cumplir en lo posible y de forma simultánea las tres siguientes restricciones:

---

**Algorithm 6** Pseudo-código del procedimiento de búsqueda local en un individuo con eliminación y/o desplazamiento de estaciones base

---

**Require:** An individual

**Ensure:** A similar feasible individual with same or better fitness value

```

1: Generate the  $N \times M$  coverage matrix
2: for each station of the individual do
3:   if less coverage than  $\beta\%$  then
4:     Remove station
5:   end if
6: end for
7: Update the  $N \times M$  coverage matrix
8: for each  $i$  station in random order do
9:   if number of locations fulfilled by only the  $i$  station  $< \beta\%$  then
10:    Check  $i$  station as expendable
11:    switch (do with expendable  $i$  station)
12:      case Deleting:
13:        Remove  $i$  station
14:      case Moving:
15:        Move station to other random and free location in  $A$ 
16:    end switch
17:    Update the  $N \times M$  coverage matrix
18:   end if
19: end for

```

---

1. Minimizar el incremento total de campo eléctrico  $\Delta E_T$  (V/m) en el área de estudio  $A$  debido al funcionamiento de las  $N$  estaciones base.
2. Minimizar el coste económico total  $\mathcal{C}_{\mathcal{T}}$  (euros) de la instalación de las  $N$  estaciones base que componen la red de telecomunicación.
3. Maximizar el grado de cobertura de la red, o lo que es lo mismo, minimizar el porcentaje de localizaciones que quedan sin servicio  $P_{no\ cober}$ , fuera de la zona de cobertura de cualquiera de las  $N$  estaciones.

En la Ecuación 4.6 se formula matemáticamente la función de evaluación atendiendo a las tres restricciones anteriores. El objetivo es por tanto minimizar el valor de esta función, también llamada función de *fitness*. El cálculo de los parámetros  $\Delta E_T$ ,  $\mathcal{C}_{\mathcal{T}}$  y  $P_{no\ cober}$  se realiza como explicamos en el Capítulo 3.

$$\mathcal{F} = k_1 \cdot \Delta E_T + k_2 \cdot \mathcal{C}_{\mathcal{T}} + k_3 \cdot P_{no\ cober} \quad (4.6)$$

Puesto que cada restricción se corresponde con una magnitud diferente, precisamos de los coeficientes  $k_1$ ,  $k_2$  y  $k_3$  para realizar la adición de términos adimensionales. Las unidades de los coeficientes son las inversas a las del parámetro que multiplican en la Ecuación 4.6. Según el valor dichos coeficientes, cada una de las tres restricciones tendrán mayor o menor peso en la función objetivo. Es necesario ajustar previamente el valor de estos coeficientes, asegurando así el adecuado peso o importancia requerida a cada una de las tres magnitudes descritas en las restricciones del problema. Para las simulaciones realizadas en el Capítulo 6, explicamos el método de ajuste de los coeficientes  $k_1$ ,  $k_2$  y  $k_3$ .

### 4.1.9 Condiciones del parada

El EA evoluciona generación tras generación según el diagrama de flujo de la Figura 2.2 hasta satisfacer el criterio de parada que le permita salir del bucle y no ejecutar más generaciones. El criterio de salida se cumple cuando se hace verdadera al menos una de las dos siguientes condiciones:

- Finalizar tras la ejecución de número máximo de generaciones  $G_{max}$ .
- Alcanzar cierto grado de convergencia en la población.

El número máximo de generaciones  $G_{max}$  que va a ejecutar el algoritmo es un parámetro que se establece al comienzo de la ejecución. Tras finalizar el proceso de la última generación  $G_{max}$ , el algoritmo ofrece la mejor solución encontrada hasta el momento y concluye su actividad. El valor de  $G_{max}$  suele establecerse de forma experimental, determinando un umbral mínimo de número de generaciones a partir del cual no compense el tiempo extra de ejecución con la mejoría en calidad de la solución que pudiera obtenerse.

Definimos grado de convergencia en la población como la diferencia entre el valor de la función objetivo o valor de *fitness* del mejor individuo  $\mathcal{F}_{best}$  y el valor medio de la función objetivo para el conjunto de individuos que componen dicha población  $\bar{\mathcal{F}}$ . La Ecuación 4.7 expresa matemáticamente la segunda condición de parada del EA, donde el grado de convergencia es menor que un valor  $\epsilon$  establecido. En el Capítulo 6 se indican los valores para  $G_{max}$  y  $\epsilon$  utilizados en las simulaciones realizadas para los resultados que se presentan en dicho capítulo.

$$\bar{\mathcal{F}} - \epsilon \leq \mathcal{F}_{best} \leq \bar{\mathcal{F}} + \epsilon \quad (4.7)$$

## 4.2 Algoritmo *Greedy* desarrollado para el problema MNDP

Un algoritmo *greedy*, conocido también como algoritmo “voraz”, realiza una búsqueda heurística que consiste en elegir la opción óptima en cada paso de su ejecución. Es decir, en cada toma de decisión se elige la mejor opción en función de la información *local* disponible. En la gran mayoría de los casos, esto no conduce a la solución óptima (*óptima global*) del problema de optimización. Un algoritmo *greedy* tiene en general una implementación sencilla pero no suelen obtener los mejores resultados. Utilizamos en este trabajo el algoritmo *greedy* con el objetivo de establecer una cota inferior para los resultados del resto de los algoritmos implementados.

Otra característica añadida de un algoritmo *greedy* es que una vez acatada una decisión, ésta no vuelve a replantearse en el futuro. Como consecuencia de ello, no pueden corregirse los errores cometidos en el pasado ni reconducir el camino de búsqueda hacia una solución mejor. Este tipo de algoritmos identifican los siguientes elementos del problema a resolver:

- Conjunto de soluciones candidatas al problema. El algoritmo finaliza cuando termina de inspeccionar completamente el conjunto de candidatos.
- Función solución, que define si un candidato es en sí mismo una solución del problema o no. En nuestro caso, esta la función solución es la propia codificación para el problema MNDP que describimos en la Sección 4.1.1, en la página 55.
- Función objetivo o función *fitness* que evalúa el grado de proximidad de una solución a la óptima. Dado que deseamos realizar la comparación de varios algoritmos, todos ellos deben utilizar la misma función de evaluación.

- Función de selección para determinar qué candidato es el más adecuado de acuerdo a la información local disponible. En el problema MNDP, esta función es la que minimiza la función de evaluación o función de *fitness*.
- Función de factibilidad que se desglosa en dos procesos: la operación de búsqueda local y la función de optimización que estudia si es posible mejorar progresivamente el individuo para alcanzar una solución de mayor calidad. El algoritmo de búsqueda local es el mismo que en el algoritmo evolutivo, detallado en la Sección 4.1.7. La función de optimización la explicamos a continuación.

El algoritmo comienza generando un individuo aleatorio de forma idéntica a como se explicó para el algoritmo evolutivo en la Sección 4.1.2. Para este individuo solución se estudia si es posible mejorarlo retirando alguna de sus estaciones base. El pseudo-código del Algoritmo 7 describe la estrategia de optimización llevada a cabo sobre el individuo. Se revisan todas y cada una de sus estaciones una sola vez y en orden aleatorio. Se estudia si retirando la estación base el valor de *fitness* del individuo se aproxima al óptimo. En caso afirmativo se elimina definitivamente la estación base de la solución. En caso negativo no se modifica nada y el análisis prosigue con la siguiente estación base.

---

**Algorithm 7** Pseudo-código general para el algoritmo *Greedy*

---

**Require:** Initial random deployment  $D$  solution with  $N = N_{MaxB}$  base stations in random locations.

**Ensure:** Feasible deployment  $D$

- 1: Initial local searching process.
  - 2: Compute fitness function  $f(D)$
  - 3: **for** each station  $i$  in random order **do**
  - 4:   Remove  $i$  from  $D$ , obtaining  $D'$
  - 5:   Compute fitness function of  $f(D')$
  - 6:   **if**  $f(D') < f(D)$  **then**
  - 7:      $D = D'$
  - 8:   **end if**
  - 9: **end for**
- 

### 4.3 Algoritmo PSO desarrollado para el problema MNDP

En esta sección describimos brevemente las particularidades del algoritmo PSO implementado para la resolución del problema de optimización. Las partes funcionales específicas de este algoritmo son: en primer lugar, la adaptación del algoritmo con ecuaciones en el espacio real  $\mathbb{R}$ , al espacio de soluciones discretas, y en segundo lugar, la adaptación de la codificación empleada en el problema MNDP al algoritmo PSO.

En la primera iteración del PSO se genera la población o *swarm* de  $N_{ind}$  partículas que convivirán durante toda la evolución del algoritmo. Estas partículas son aleatorias, tanto en el número de estaciones base que contiene, como en la localización y tipología de cada una de sus estaciones. Tal y como describimos en la Sección 4.1.2 para el EA, cada partícula contiene un número aleatorio de estaciones base comprendido en el intervalo  $[N_{MinB}, N_{MaxB}]$ . Dicho intervalo se determina mediante el método indicado en el EA.

Respecto al valor de  $N_{ind}$ , en [8] se indica que el tamaño del *swarm* influye en el rendimiento del algoritmo de forma variable y dependiendo del problema de optimización. Algunas funciones

de prueba experimentan ligeras mejorías cuando se incrementa  $N_{ind}$ , mientras otras tienden hacia mejores soluciones cuando este parámetro se reduce. Parece no existir un valor de  $N_{ind}$  que resulte idóneo para cualquier problema de optimización. [8] sugieren valores comprendidos entre 20 y 100 partículas ya que las pruebas realizadas no ofrecen una variación significativa en los resultados para este rango de  $N_{ind}$ .

Por otro lado, en [97] se afirma que el PSO no se muestra notablemente sensible hacia este parámetro. Los autores de [12] aclaran que esto es generalmente cierto en términos de resultados, pero no en términos de coste computacional. Proponen un valor para  $N_{ind}$  comprendido entre 25 y 30 como solución de compromiso. En el Capítulo de 6 incluimos simulaciones con *swarms* de 25 y 50 partículas, comprobando que efectivamente los resultados no varían, al contrario de la diferencia de carga computacional.

En cuanto al criterio de parada del algoritmo es válido aquí todo lo mencionado para el EA en la Sección 4.1.9. En general la ejecución finaliza cuando se alcanza un número máximo de iteraciones  $G_{max}$ , entonces el algoritmo entrega la mejor posición encontrada entre todas las partículas durante toda la ejecución.

#### 4.3.1 Conversión del espacio de soluciones continuo al discreto

El problema de optimización MNDP es un problema con solución en el espacio discreto, donde cualquier solución está compuesta de localizaciones de estaciones base y tipos de estas, tomando ambos parámetros siempre valores discretos. El modelo clásico del PSO propuesto por [63] está desarrollado para resolver problemas en entornos discretos. Sin embargo, las Ecuaciones 2.8 y 2.9 están definidas en espacios continuos donde tanto la posición como la velocidad de la partícula quedan determinadas por combinación lineal de varios vectores. La Ecuación 2.9 calcula el vector velocidad en la iteración  $(k+1)$  como la combinación lineal de otros tres: velocidad en la iteración  $k$ , componente cognitiva y componente social. La Ecuación 2.8 expresa el vector posición de la partícula en la iteración  $(k+1)$  como la suma de otros dos vectores, el de posición en la iteración  $k$  y el vector velocidad en la iteración  $(k+1)$ . Podemos decir por tanto, que el algoritmo PSO se encuentra desarrollado para problemas de optimización en el espacio real  $\mathbb{R}$ .

Muchos investigadores, entre ellos Kennedy y Eberhart, han creado versiones de este método con el objetivo de aplicarlo en problemas discretos. Sin embargo, no existe un modelo de optimización de PSO que pueda ser aplicado a todos los problemas de tipo discreto de forma general. Los algoritmos propuestos contienen características muy específicas del problema sobre el que se aplica. En la búsqueda por encontrar un método más completo para aplicar PSO en cualquier problema combinatorio, se ha investigado en [53] cuatro diferentes enfoques de lo que se denominan algoritmos de PSO discretos (DPSO):

1. Basados en operadores de intercambio. El algoritmo define la posición de la partícula como una permutación de números y la velocidad como un conjunto de intercambios [19, 103].
2. Transformación del espacio. La posición es definida como un vector real y una técnica de transformación del espacio es usada para convertir la posición en su correspondiente solución [81, 91, 92].
3. Matrices difusas. Estos métodos definen a la posición y a la velocidad como matrices difusas. Se necesita un operador que, en el dominio discreto, decodifique la matriz difusa en una solución posible [69, 80, 93].
4. Algoritmos híbridos. Se trata de cualquier técnica metaheurística añadida al PSO [2, 104], o utilice una búsqueda local [15, 59, 79, 94, 101] con el objeto de mejorar la solución.

Esta estrategia, llamada hibridación del método, se aplica en alguna etapa del algoritmo principal. En las referencias consultadas la mayoría de los esfuerzos están dirigidos a emplear PSO en problemas discretos con técnicas de búsqueda local incorporadas.

Los tres primeros tipos de algoritmos PSO discretos siguen una estructura basada casi en su totalidad en la concepción clásica de PSO. Sin embargo, comparados con otros algoritmos metaheurísticos para optimizaciones discretas los resultados no son competitivos. Por ejemplo, en el *problema del viajante* (*Travelling Salesman Problem, TSP*), los mejores resultados obtenidos por los tres primeros tipos de algoritmos DPSO están aún lejos de los obtenidos por algoritmos de optimización basadas en colonias de hormigas (*Ant Colony Optimization, ACO*).

Por otro lado, los algoritmos híbridos DPSO funcionan mucho mejor que los otros enfoques DPSO más clásicos. Estos algoritmos híbridos se diseñan generalmente para problemas específicos, sus estructuras son muy complicadas y no resulta sencillo aplicarlos a otros problemas de optimización combinatoria.

En nuestro trabajo hemos optado por diseñar un DPSO de tipo híbrido adaptado al problema MNDP y apoyado en la búsqueda local descrita en la Sección 4.1.7 del EA. La búsqueda local permite por un lado mejorar el valor *fitness* de las soluciones encontradas variando la posición de estaciones base prescindibles o bien eliminándolas de la solución.

Realizamos una conversión de los valores de velocidad a valores discretos después de actualizar la posición de las partículas (Ecuaciones 2.8 y 2.10). En [41] se aplica un DPSO que utiliza un método muy similar al propuesto en este estudio para trabajar con valores discretos. El artículo convierte todas las variables continuas de la ecuación velocidad a discretas, luego calcula la velocidad usando estas variables discretas, actualiza las posiciones y vuelve a pasarlas a un entorno discreto.

### 4.3.2 Codificación empleada

La codificación utilizada en el algoritmo evolutivo y expuesta en la Sección 4.1.1, representa cada solución como un conjunto de  $N$  pares  $(p_i|t_i)$  donde  $p_i$  identifica la localización de la estación base  $i$  con  $p_i \in \mathcal{P}$ , mientras que  $t_i$  indica el tipo estación base instalada en dicha localización, con  $t_i \in \mathcal{T}$ .

$$c_{EA} = \{(p_i|t_i)\}, i = 1, \dots, N$$

En el algoritmo PSO desarrollado, cada partícula o solución compuesta por  $N$  estaciones base se encuentra representada por un vector  $c$  en el que cada una de sus  $N$  componentes toman como valor un número entero que identifica completamente a una estación base, tanto la ubicación como el tipo al que pertenece.  $\#\mathcal{P}$  representa el cardinal del conjunto  $\mathcal{P}$ , es decir, el número de localizaciones posibles para la instalación de estaciones base en la zona de estudio  $A$ . De forma análoga,  $\#\mathcal{T}$  es el cardinal del conjunto  $\mathcal{T}$  o número de tipos diferentes de estaciones base disponibles para el despliegue de red.

$$c = \{(x_i)\}, i = 1, \dots, N$$

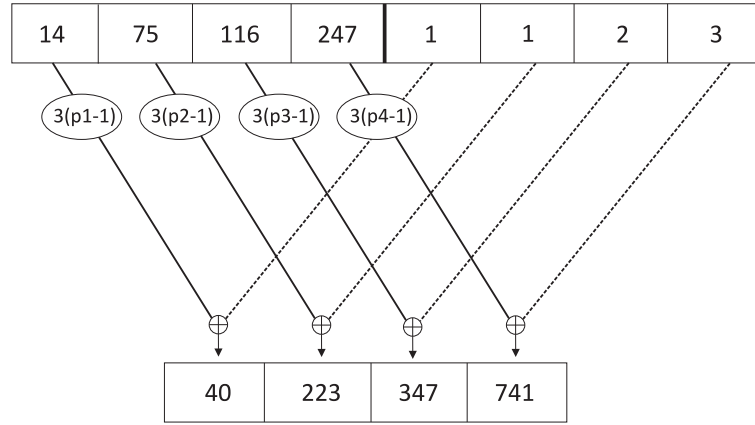
$$x_i \in [1 \dots, (\#\mathcal{P} \times \#\mathcal{T})]$$

La transformación de cada pareja de  $(p_i|t_i)$  en una única componente  $(x_i)$  se realiza aplicando la Ecuación 4.8. La Figura 4.9 muestra un ejemplo de codificación de una solución compuesta por cuatro estaciones base. Las localizaciones de las estaciones tienen los identificadores 14, 75, 116 y 247. Sus tipos son 1, 1, 2 y 3, respectivamente. El primer vector de la figura, cuya



longitud es 4+4, corresponde al individuo codificado por el EA, mientras que el vector de cuatro componentes es el vector que se corresponde con la partícula solución codificada por el algoritmo PSO. Para realizar la transformación de un vector a otro según la Ecuación 4.8 necesitamos conocer el número de tipos posibles de estaciones base, que en este caso es  $\#\mathcal{T} = 3$ . Hemos considerado que el número total de localizaciones es  $\#\mathcal{P} = 509$ . De este modo, los valores para la partícula serán números enteros en el intervalo [1-1527]. La figura indica la operación realizada con cada par  $(p_i|t_i)$  obteniendo su correspondiente  $x_i = 3(p_i - 1) + t_i$ .

$$x_i = (\#\mathcal{T})(p_i - 1) + t_i \quad (4.8)$$



**Figura 4.9:** Ejemplo de codificación para el algoritmo PSO de una solución al problema MNDP a partir de la codificación empleada en el algoritmo evolutivo.

## 4.4 Algoritmo HS desarrollado para el problema MNDP

El algoritmo HS implementado para la resolución del problema MNDP emplea al igual que los algoritmos descritos anteriormente algunas funciones ya implementadas para el algoritmo evolutivo. En concreto utiliza el mismo método de codificación para las soluciones y la misma función de evaluación para calcular el valor de *fitness* de cada solución. Además, cuando el nuevo acorde o solución debe ser generado mediante la acción *randomization*, se realizan las mismas operaciones que en el algoritmo evolutivo: selección de un número aleatorio de estaciones base dentro de un rango predefinido según la extensión del área de estudio y las características de las estaciones base disponibles. En la Sección 4.1.2 ya explicamos el modo de determinar este intervalo.

El algoritmo HS no incluye ninguna operación de búsqueda local. A cambio de ello se ha dotado a la operación de *pitch adjustment* de la posibilidad de añadir a la solución nuevas estaciones base aleatorias y/o eliminar alguna/s de ellas. En el Capítulo 6 se indica la probabilidad de esta acción.

La condición de parada del algoritmo, tal y como se explicó en el Capítulo 2 consiste en alcanzar un número determinado de iteraciones del algoritmo. Tal y como se explicó en este Capítulo se trata de un algoritmo de implementación sencilla con resultados relativamente buenos. El Algoritmo 8 recoge el pseudo-código del HS. Tras inicializar una memoria HM con soluciones aleatorias y establecidos los valores deseados para los parámetros característicos del algoritmo (HMS, HMCR, PAR y *bw*) se ejecuta un bucle de iteraciones en busca de soluciones o acordes

con mejor calidad que las registradas en la memoria. Recordamos las tres acciones posibles para componer un nuevo acorde o solución en cada una de las interacciones: *memory consideration*, *pitch adjustment* y *randomization*. Estas acciones se llevan a cabo respetando las probabilidades HMCR y PAR de la siguiente manera, tal y como se describió en el Capítulo 2:

- Utilización del conocimiento de la HM con probabilidad HMCR.
  - Acción de *memory consideration* con probabilidad  $HMCR \times (1 - PAR)$ .
  - Acción de *pitch adjustment* con probabilidad  $HMCR \times PAR$ .
- Operación de aleatoriedad o *randomization* con probabilidad  $(1-HMCR)$ .

---

**Algorithm 8** Pseudo-código general para el algoritmo HS
 

---

**Require:** Initial random harmonics in HM

**Ensure:** An harmonics with a good fitness corresponding to a feasible deployment  $D$

```

1: while iteration < Max. number of iterations do
2:   if rand() > HMCR then
3:     generate a new harmonics via memory consideration
4:     if rand() > PAR then
5:       adjust the pitch randomly within limits
6:     end if
7:   else
8:     generate new harmonics via randomization
9:   end if
10:  if the new harmonics is better than the worst one in HM then
11:    the new harmonics overwrites the worst one in HM
12:  end if
13: end while
14: Return the best harmonics (solution)

```

---

El manejo de la HM precisa una explicación particularizada para el problema MNDP. Como ya comentamos en el caso del algoritmo evolutivo, la codificación empleada permite soluciones de longitud variable. Precisamente esta característica obliga a que la operación de *memory consideration* resulte algo más compleja que lo explicado en el Capítulo 2. En la Figura 4.10 se muestra un ejemplo de HM compuesta por cuatro soluciones. Hemos destacado en cada una de ellas la mitad izquierda conteniendo información de las localizaciones de las estaciones base y la mitad derecha que registra la tipología de las mismas. Es importante hacer notar que todas las soluciones ordenan sus estaciones base de forma que sus localizaciones aparecen siempre en orden creciente. El ejemplo es una memoria con cuatro soluciones con longitud variable: seis estaciones base la primera de ellas, cuatro la siguiente y nueve estaciones base las dos últimas soluciones de la HS.

A la hora de generar una nueva solución a partir de la HM mediante la operación de *memory consideration*, es necesario tener en cuenta algunas particularidades. La Figura 4.11 desarrolla el proceso de la operación de *memory considering* para la HM de la Figura 4.10. En primer lugar expone la estructura de la nueva matriz HM adaptada a la codificación empleada. Recordemos que el elemento  $x_j^i$  de la matriz HM se localiza en la fila  $i$  y en columna  $j$ . En el símil de la orquesta de jazz, su contenido es precisamente la nota del acorde  $i$  para el instrumento musical  $j$ . En el problema de optimización  $x_j^i$  es el valor que indica la solución número  $i$  para la variable de decisión  $j$ . En esta nueva estructura,  $x_j^i$  contiene toda la información relativa a una estación

Harmonics in HM:

monics in AIV:

1	10	12	24	33	44	3	1	1	3	2	2						
2	18	24	32	1	3	2	1										
1	2	10	12	18	24	32	33	44	3	1	1	3	3	2	1	2	2
1	12	14	18	19	32	44	52	76	2	1	1	1	1	3	2	3	1

Figura 4.10: Ejemplo del contenido de la *harmony memory* (HM) para el problema MNDP.

base (tanto su ubicación como sus características de frecuencia, potencia, coste, capacidad, etc.). El elemento  $x_j^i$  se compone de dos identificadores dispuestos en celdas verticales. Por ejemplo, el elemento  $x_4^2 = \{32, 1\}$  describe la cuarta estación base de la segunda solución de la HM. Dicha estación se encuentra ubicada en la posición 32 del conjunto  $\mathcal{P}$  y es de tipo 1 dentro del conjunto  $\mathcal{T}$ . Podemos observar que en la estructura quedan elementos  $x_j^i$  sin contenido, debido a que no todas las soluciones tienen el mismo número de estaciones base.

HM structure:

$x_i^i$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$
$x^1$	1	10	12	24	33	44			
$x^2$	2	18	24	32					
$x^3$	1	2	10	12	18	24	32	33	44
$x^4$	1	12	14	18	19	32	44	52	76

Memory consideration:

	i=3	i=4	i=3	i=2	i=4	i=2	i=3	i=1	i=4
$x'$	1	12	10	32	19	32	76		
	3	1	1	1	1	1	1		

Ordering:

$x'$	1	10	12	19	32	76
	3	1	1	1	1	1

Figura 4.11: Ejemplo de la operación de *memory consideration* a partir de la HM de la Figura 4.10.

Supongamos ahora que se aplica la operación de *memory consideration* con probabilidad

HMCR=1. Con esta probabilidad tratamos de simplificar el ejemplo anulando la posibilidad de *randomization*. Supongamos también que la operación de *memory consideration* genera la secuencia aleatoria de números de filas  $i = \{3, 4, 3, 2, 4, 2, 3, 1, 4\}$ . La Figura 4.11 muestra esta secuencia para construir la nueva solución  $x'$ . La longitud de la secuencia coincide con el número máximo de estaciones base de todas las soluciones almacenadas en la HM, en este caso es nueve. Sin embargo no se garantiza que la nueva solución  $x'$  también tenga nueve estaciones base. De hecho, el ejemplo de la figura muestra dos ocasiones, elementos rayados en la Figura 4.11, en las que no existe estación base en la estructura de la matriz. Estas dos ocasiones hacen que la nueva solución tenga dos estaciones base menos que la solución de mayor longitud en la memoria, quedando por tanto con siete estaciones base pendientes de ordenar de forma creciente por su identificador de localización. La Figura 4.11 efectúa esta acción (*ordering*) encontrando dos estaciones ubicadas en la localización 32 y eliminando una de ellas al azar.

La acción de *memory consideration* no asegura que la nueva solución tenga un número mínimo de estaciones base, puesto que elimina las que resultan coincidentes en localización. El número de estaciones base que puede alcanzar una solución mediante esta operación es la longitud máxima de las soluciones de la HM.

Construida la nueva solución o acorde  $x'$ , se evalúa y compara con la peor de las almacenadas en la memoria. En caso de mejorar el valor de *fitness* de alguna de ellas, la nueva solución ocupa el lugar de la peor de las contenidas en la HM antes de proseguir con la siguiente iteración. El resultado final es la solución con mejor calidad encontrada durante toda la ejecución.

## Capítulo 5

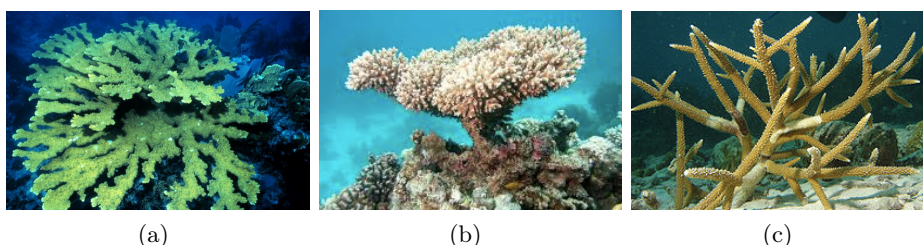
# Algoritmo Coral Reefs Optimization

En este capítulo describimos el algoritmo desarrollado en el trabajo de tesis como un nuevo método heurístico para la resolución del problema de optimización MNDP. Se trata del algoritmo *Coral Reefs Optimization* (CRO), recientemente propuesto en [88, 89]. Este algoritmo está basado en el comportamiento de los corales en el ecosistema marino. Se trata de un algoritmo de tipo evolutivo en el que cada individuo representa una solución al problema. A través de sucesivas generaciones, los individuos evolucionan convirtiéndose en otros que representan soluciones cada vez más próximas a la óptima. La evolución del sistema en este caso se basa en los procesos de formación de un arrecife de coral y lucha por el espacio físico en él.

En la primera sección, describimos las características básicas de los arrecifes de coral en la naturaleza, características que luego resultarán puntos importantes para introducir el algoritmo CRO. A continuación describimos la funcionalidad del algoritmo CRO y su aplicación en el problema MNDP, para terminar el capítulo con los detalles de la implementación del mismo.

### 5.1 Arrecifes de coral en la naturaleza

Hasta el siglo XVIII se creía que los corales eran plantas, el científico William Herschel los identificó como animales marinos al descubrir bajo microscopio que los corales tienen membranas celulares de tipo animal. Aunque una colonia de coral parece un único organismo, realmente está compuesto por muchos organismos multicelulares independientes denominados pólipos o corales que se adhieren a un sustrato. Los corales pertenecen a la misma categoría que las anémonas de mar y que las medusas [11]. Las anémonas de mar, a diferencia de los corales, suelen encontrarse aisladas sin formar colonias y son mayores que los corales. A modo de ejemplo, la Figura 5.1 muestra tres ejemplares de diferentes especies de corales.



**Figura 5.1:** Fotografías de diferentes corales en la naturaleza obtenidas a través de la revista National Geographic.

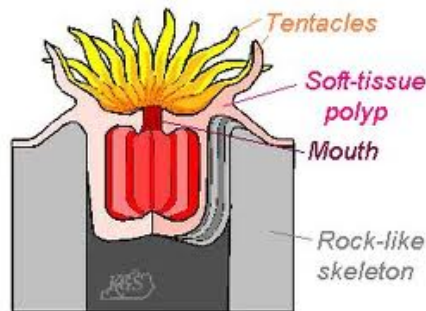
Los arrecifes de coral constituyen parte de los ecosistemas más ricos y biodiversos de la tierra,

florecen a pesar de encontrarse rodeados por aguas oceánicas tropicales que apenas proporcionan nutrientes. Cubriendo menos del 0,1% de la superficie de los océanos del mundo, los arrecifes de coral sustentan a más de una cuarta parte de todas las especies marinas, desde los propios corales, algas, esponjas, crustáceos, moluscos hasta una multitud de peces. En este ecosistema la cadena alimenticia comienza con los grandes peces depredadores de otros menores, éstos a otros bajando en la cadena hasta los pequeños peces forrajeros, que comen los aún más pequeños zooplancton, y así sucesivamente.

Según el esqueleto de los corales, éstos se clasifican en dos grandes grupos: corales duros o hermatípicos y corales blandos o ahermatípicos. Los corales hermatípicos, son corales pétreos que construyen arrecifes y de los que hablaremos fundamentalmente en este trabajo.

Los corales se clasifican también según la estructura general de la colonia. Así, por ejemplo, la familia *Fungiidae* son los corales de aspecto similar a los hongos, los corales *Tubiporidae* destacan por su semejanza a tubos de órgano y los corales *Milleporidae* tienen apariencia de llamas encendidas bajo las aguas.

La Figura 5.2 es la representación esquemática de un pólipo o coral. Éstos son animales muy sencillos. Están formados por una capa externa o *epitelio* y otra capa de tejido interno o *mesoglea*. Los corales tienen simetría radial, pueden medir desde varios milímetros hasta algunos centímetros de diámetro. Se componen de varios tentáculos que rodean una boca central que comunica directamente con un estómago o *celenteron*. Cuando un coral atrapa una presa, éste se contrae para retenerla en su estómago. Una vez digerido el alimento, el estómago se abre para expulsar los residuos al exterior a través de la boca. El estómago se cierra en la base donde el coral genera un exoesqueleto denominado placa basal o cálculo. El cálculo está formado por un anillo calcáreo que crece verticalmente por extensión de cálices. Con el paso de muchas generaciones, estas estructuras calcáreas forman los conocidos arrecifes de coral. Cuando el coral sufre algún tipo de peligro, sus tentáculos se contraen y se protege escondiéndose dentro de su exoesqueleto [6, 99].



**Figura 5.2:** Esquema básico de la estructura de un pólipo de coral. Figura realizada por el Fulton-Montgomery Community College (<http://faculty.fmcc.suny.edu/>).

La estructura calcárea del coral es de color blanco. Los variados colores que exhiben los corales se deben a diferentes pigmentos de sus tejidos y a las zooxantelas. Las zooxantelas son microalgas que viven en simbiosis con los pólipos coralinos que habitan en aguas transparentes, donde las zooxantelas pueden realizar la fotosíntesis. Típicamente, cada coral alberga una especie de alga. A través de la fotosíntesis, las zooxantelas proporcionan energía al coral y le ayudan en el proceso de calcificación. Estas algas producen oxígeno y azúcares que aprovechan los corales, mientras que las algas se alimentan de los catabolitos del coral (especialmente fósforo y nitrógeno). Las zooxantelas proporcionan a los corales alrededor del 80% de sus necesidades

alimenticias. Estos últimos completan su dieta a base del plancton que atrapan. Cuando el coral muere se produce el fenómeno conocido como *blanqueo del coral* donde el coral pierde el color vistoso que lo caracteriza.

La forma de reproducción de los corales difiere según su tipo [11]. Incluso a veces pueden reproducirse dos corales de diferentes especies, siempre que sean especies compatibles. En general podemos hablar de cuatro tipos de reproducción: asexual por gemación, asexual por división, sexual externa y sexual interna.

La reproducción asexual por gemación (*budding*) consiste en dividir un coral adulto en dos porciones de tamaño desigual [107]. A partir de ahí, los dos pedazos se desarrollan hasta completar sus partes. La división del coral en el caso de la reproducción asexual por división (*fragmentation*) está causada por factores externos como tormentas o colisiones con embarcaciones. La fragmentación puede ser bien longitudinal o bien transversal [70]. El animal es capaz de regenerar las partes que le falta en cada fragmento. En ambos tipos de reproducción asexual se generan individuos genéticamente idénticos. La reproducción asexual tiene numerosos y beneficiosos efectos para los corales: elevada tasa de reproducción, rápida explotación del espacio del arrecife, retraso de su envejecimiento puesto que puede sobrevivir a la pérdida de masa, reducción de la mortalidad puesto que los nuevos corales pueden ocupar los espacios que dejaron libres otros y expansión de los clones a nuevas zonas perdurando la especie a pesar de adversidades.

La reproducción sexual externa o también denominada reproducción por difusión o *broadcast spawning* consiste en la expulsión al exterior de gametos masculinos y femeninos. Este método es el más común, dándose en el 75% de los corales hermatípicos [75]. Los gametos se fusionan en el agua para generar larvas de tamaño microscópico o *plánulas*. En algunas especies, un único coral es capaz de producir gametos de ambos sexos. Existe un especial sincronismo en el desove de los arrecifes de coral. Dicho sincronismo está dirigido por señales ambientales como cambios de temperatura, ciclo lunar, marea baja, duración del día o comienzo de la puesta de sol. Cuando diferentes especies de coral se someten al mismo sincronismo surgen fecundaciones híbridas, una tercera parte de estas suelen ser compatibles fomentando la especialización y evolución del coral. Cada una de las plánulas nada hacia la luz (*fototaxia positiva*) y alejándose del mar abierto (*sonotaxia positiva*) y crece durante unos tres días. Entonces comienza a descender para buscar un estrato donde establecerse [100]. En su desarrollo sufre transformaciones, pierde el *cilio* que le servía para nadar y se forman la boca y los tentáculos que usará en el futuro para alimentarse. Aunque con probabilidad reducida, un único individuo de coral podría originar la construcción de una nueva colonia mediante reproducción asexual en los primeros ciclos. Un arrecife produce al año miles de larvas. El índice de fracaso es muy elevado debido a multitud de circunstancias desfavorables: falta de condiciones adecuadas para la fecundación, depredación, oleaje, etc.

La reproducción sexual interna o incubación (*brooding*) es utilizada por corales no hermatípicos. Aquí también participan gametos masculinos y femeninos. Los masculinos son liberados al agua como en la reproducción sexual externa. Sin embargo los gametos femeninos permanecen dentro del individuo. Los masculinos carecen de flotabilidad y se hunden tratando de encontrar su oportunidad de fecundar un gameto femenino dentro de un coral. Tras la fecundación, el coral portador libera la nueva plánula en avanzado estado de desarrollo, aumentando así las probabilidades de supervivencia del nuevo individuo [9].

## 5.2 Estrategia del algoritmo Coral Reefs Optimization

Tras la breve exposición acerca de los arrecifes de coral en la naturaleza y su funcionamiento desde el punto de vista biológico, presentamos el algoritmo de optimización CRO. Se trata de un algoritmo bio-inspirado que modela la forma de vida de las colonias de coral. En el algoritmo,

cada individuo representa una solución posible del problema de optimización a resolver. En cada generación los corales realizan las funciones de crecimiento, reproducción y lucha por su supervivencia. Compiten entre ellos por continuar asentados en la estructura de arrecife. Los individuos mejor adaptados corresponden a las soluciones más próximas a la solución óptima y por lo tanto, con mejor valor de la función de evaluación. El principal problema de estos animales, aparte de obtener su alimento, es encontrar un lugar en la estructura de arrecife donde asentarse y mantener su propiedad. El espacio en el arrecife es un recurso muy limitado [48] comparado con el altísimo porcentaje de natalidad de estos animales. Surge entonces la competición que en algunas especies destaca por su agresividad [5, 13]. Como resultado de esta contienda parte de los individuos mueren al no poder defender el espacio conquistado o no encontrar un espacio donde ubicarse. Tras la ejecución de un número determinado de generaciones en el arrecife, el algoritmo extrae la solución con mejor valor de *fitness*. El CRO es un algoritmo de tipo evolutivo en el que generación tras generación, los individuos que habitan en la estructura tienen en media mejor valor de la función de evaluación.

En el algoritmo CRO, todas las operaciones de interacción entre individuos, los métodos de reproducción y los mecanismos de supervivencia están basados en el comportamiento natural de los arrecifes de coral. En los siguientes apartados se describen de manera general los parámetros característicos del algoritmo CRO, su funcionamiento básico y los métodos reproductivos implementados.

### 5.2.1 Parámetros específicos del algoritmo CRO

El problema de optimización MNDP se caracteriza por un conjunto de parámetros que aparecen de una forma u otra en todos los algoritmos desarrollados en este trabajo. Algunos de estos son: los conjuntos  $\mathcal{P}$  y  $\mathcal{T}$  que identifican posibles localizaciones y tipos de estaciones base, la codificación básica de una solución compuesta por  $N$  estaciones base, el establecimiento del rango de número de estaciones base para individuos de nueva creación o la propia función de evaluación. El algoritmo CRO utiliza además un conjunto de parámetros específicos, de los cuales, los más representativos se recogen a continuación:

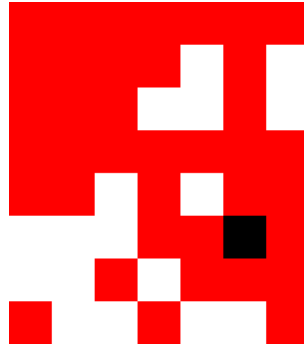
- Capacidad máxima del arrecife: la estructura puede albergar hasta un máximo de  $N_{ind}$  corales o individuos. El algoritmo define el arrecife como una estructura de celdas bidimensional con forma rectangular. Sus dimensiones son  $\mathcal{N} \times \mathcal{M}$ , donde  $N_{ind} = \mathcal{N} \times \mathcal{M}$ .
- Factor de ocupación inicial  $\rho$ : determina el número de corales existentes en el arrecife en el instante de comienzo del algoritmo.
- Probabilidad de reproducción sexual externa ( $Pb_{ext}$ ): esta probabilidad establece el porcentaje de corales que llevarán a cabo este método de reproducción en una generación. Los corales son elegidos aleatoriamente para una u otra forma de reproducción.
- Probabilidad de reproducción sexual interna ( $Pb_{int}$ ): de forma análoga a la externa,  $Pb_{int}$  determina el porcentaje de corales que realizarán este tipo de proceso reproductivo. Dichos corales son seleccionados también de forma aleatoria de entre el conjunto de todos que no tienen reproducción sexual externa. Si la suma de las probabilidades para ambos tipos de reproducción sexual excede la unidad, entonces algunos individuos tendrán la oportunidad de reproducirse por ambos métodos. El tipo de problema y la experiencia ayudan a ajustar adecuadamente el valor de ambas probabilidades.



- Probabilidad para la reproducción asexual ( $Pb_{asex}$ ). La reproducción asexual genera una copia exacta de un coral. Para evitar la falta de diversidad en el arrecife se toman valores de  $Pb_{asex}$  inferiores al 1%.
- Porcentaje de los mejores corales ( $\%p_{best}$ ) que participan en la reproducción asexual: para evitar réplicas de individuos de baja calidad, la reproducción asexual se permite solo a los  $\%p_{best}$  corales, con mejor valor de *fitness*.
- Número máximo de intentos  $N_{Ch}$  para el asentamiento en el arrecife: los corales de nueva creación tienen un número máximo de oportunidades para establecerse en una hueco del arrecife. El asentamiento se produce cuando encuentra una posición libre o tiene mejor valor de *fitness* que el ocupante de la misma. El algoritmo resulta más elitista cuanto mayor es el valor de  $N_{Ch}$ , puesto que los individuos con peor valor de *fitness* tendían muy pocas probabilidades de mantenerse vivos.
- Porcentaje de los peores individuos  $\%p_{pred}$  participantes en el proceso de depredación. Con este parámetro, el algoritmo ofrece más posibilidades de sobrevivir a los corales con mejor valor de la función de evaluación. Ordenando a los corales de peor a mejor valor de *fitness*, envía solo los  $\%p_{pred}$  de peor calidad al proceso de depredación.
- Probabilidad de depredación  $Pb_{pred}$ : un coral que participa en el proceso de depredación es eliminado del arrecife con probabilidad  $Pb_{pred}$ .

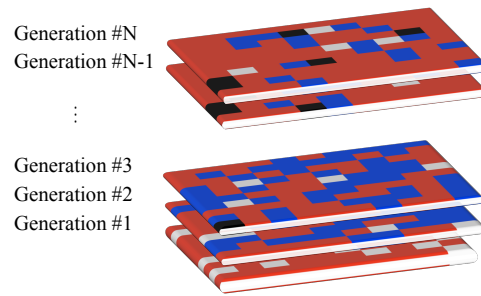
### 5.2.2 Proceso de comienzo del algoritmo

El proceso de inicio del algoritmo consiste en partir de una estructura de arrecife en el que habitan un número de corales aleatorios. Dicho número viene determinado por el factor de ocupación  $\rho$  del arrecife. Los corales iniciales aparecen dispuestos de forma aleatoria sobre la estructura de arrecife y representan cada uno de ellos una posible solución factible del problema de optimización MNDP. En la Sección 5.3.1 se explican los detalles del comienzo del algoritmo. La Figura 5.3 muestra un ejemplo de un arrecife de coral en el inicio del algoritmo CRO. Se trata de una estructura bidimensional de  $10 \times 10$  celdas, donde las celdas de color rojo indican que habita un coral, los huecos blancos representan celdas libres y la celda de color negro señala la ubicación del mejor individuo de la población. El factor de ocupación que se ha empleado en este caso es  $\rho = 70\%$ .



**Figura 5.3:** Representación gráfica del estado de un arrecife bidimensional en el instante de inicio del algoritmo CRO. El color blanco señala los huecos libres que hay en la estructura para la siguiente generación.

A través de las sucesivas generaciones del algoritmo, los corales se reproducen generando así otros nuevos que tratarán de asentarse en huecos libres del arrecife o bien deberán competir con alguno por un mismo espacio. La Figura 5.4 simboliza la evolución o crecimiento de un arrecife bidimensional a través de  $N$  generaciones. Esta evolución está inspirada en el crecimiento vertical de los corales duros en la naturaleza. El código de colores empleado es el mismo que en la Figura 5.3, donde las celdas sin ocupar quedan de color blanco, el color negro nos indica la posición para el mejor o mejores individuos de cada iteración, las celdas que toman el color azul son de nueva ocupación, es decir, en la generación  $N$  se encontraban libres y en la generación  $N+1$  quedan ocupadas. Finalmente el color rojo se utiliza para indicar que una celda se encuentra ocupada durante al menos dos generaciones consecutivas.



**Figura 5.4:** Ejemplo de la evolución de un arrecife bidimensional en el algoritmo CRO durante  $N$  generaciones.

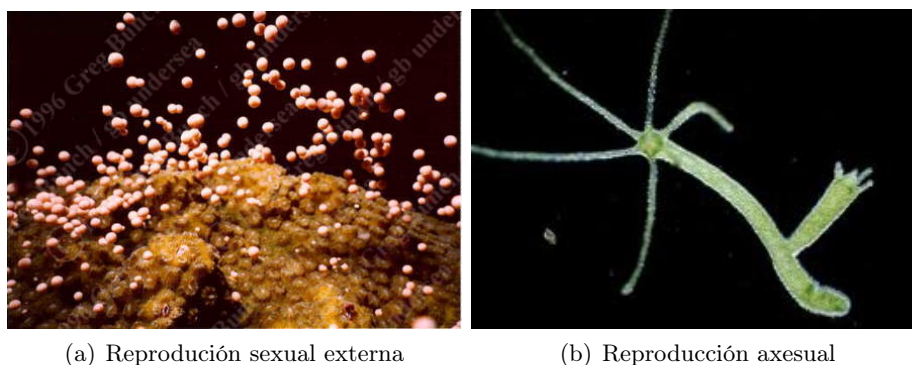
### 5.2.3 Métodos de reproducción del coral

Los métodos de reproducción de corales en el algoritmo CRO simulan algunas de las formas de reproducción de corales duros o hermatípicos en la naturaleza. Tal y como comentamos en la Sección 5.1, los corales duros tienen dos formas de reproducción claramente diferenciadas: la reproducción sexual externa y la reproducción asexual. La primera consiste en generar nuevos individuos a partir de información genética de otros dos individuos progenitores. Aunque en la naturaleza, este método tiene sus particularidades para los arrecifes de coral, este tipo de reproducción recuerda al operador de cruce o recombinación de los algoritmos genéticos. En la Figura 5.5(a) se muestra una fotografía instantánea de un coral durante la expulsión de gametos al exterior, donde surgirá un nuevo individuo a partir de cada combinación de una pareja de gametos compatibles.

Por otra parte, la reproducción asexual en los corales consiste en generar un individuo idéntico a otro. La Figura 5.5(b) ofrece una fotografía de un coral realizando este tipo de reproducción. En la parte derecha del cuerpo comienza a desarrollarse lo que acabará convirtiéndose en un nuevo coral independiente e idéntico genéticamente al progenitor. Esta función también se ha implementado en el algoritmo CRO. En la naturaleza este tipo de reproducción es muy conveniente para la especie. Sin embargo cuando el algoritmo se ejecuta sobre estructuras de pequeñas dimensiones, escasa población, se observa una tendencia a que individuos duplicados dominen la colonia. Por esta razón en el algoritmo hemos añadido cierta probabilidad de realizar pequeñas

pero significativas modificaciones sobre la réplica. Son cambios estratégicos que persiguen mejorar el valor de *fitness* del individuo.

Además de la reproducción sexual y asexual propias de los corales duros, hemos implementado en el algoritmo CRO un método de reproducción sexual interna basado en una de las formas de reproducción propias de los corales blandos. No pretende ser una analogía exacta, sino un método más utilizado en el modelo. Es similar a la reproducción asexual en el sentido de que el coral progenitor se copia a sí mismo y después se aplica una mutación. Los detalles de la implementación de los tres tipos de reproducción en el algoritmo CRO se describen en la Sección 5.3.3.



**Figura 5.5:** Fotografías instantáneas de tipos de reproducción en corales duros. Fuente de las imágenes: (a) [www.gbundersea.com](http://www.gbundersea.com); (b) <http://vcebiology.edublogs.org>.

### 5.3 Algoritmo CRO desarrollado para el problema MNDP

En el desarrollo del algoritmo para la resolución del problema de optimización MNDP empleamos nuevamente algunas de las funciones ya descritas en secciones anteriores para el resto de los algoritmos implementados. Fundamentalmente es el caso de la codificación de las soluciones del problema, la función de evaluación utilizada y la operación de búsqueda local como estrategia añadida que hace que el algoritmo sea híbrido. Todo lo explicado en el Capítulo 4 respecto a estos tres procedimientos para el EA, es aplicable ahora en el algoritmo CRO.

El algoritmo CRO realiza una simulación de la vida de una colonia de corales en un arrecife durante un número  $G_{max}$  de generaciones. En cada una se llevan a cabo los procesos de reproducción y competición por los recursos limitados en el arrecife de coral. En esta Sección explicamos detalladamente cada uno de los procesos específicos del algoritmo.

El Algoritmo 9 recoge el pseudo-código general del algoritmo CRO. Una vez inicializadas adecuadamente todas las variables del algoritmo y establecida una población inicial de corales, comienza el recorrido de éstos a través de un número finito  $G_{max}$  de generaciones o iteraciones. La condición de parada del programa es precisamente alcanzar la última generación. Momento en el cual se extrae el mejor individuo de la colonia como solución final. En cada generación se aplica la función de evaluación sobre los nuevos corales, también se realizan diferentes procesos de reproducción seguido del asentamiento de los corales vencedores en la lucha por el espacio físico en el arrecife. Hemos incorporado un proceso de depredación, como existe en la naturaleza y una función de control de réplicas para garantizar una diversidad suficiente y evitar la convergencia prematura del algoritmo. En los siguientes apartados estudiamos por separado cada uno de estos procesos y en el orden en que aparecen en el Algoritmo 9.

**Algorithm 9** Pseudo-código general del algoritmo CRO**Require:** Valid values for variables of CRO**Ensure:** A single feasible individual with good *fitness*

- 1: Initialize algorithm (see Algorithm 10)
- 2: **for** each iteration of the simulation **do**
- 3:   Update values of influential variables: predation probability, percentage of candidates for predation and the probability of asexual crossover
- 4:   Control the number of copies in population (see Algorithm 11)
- 5:   Sexual crossover process (see Algorithm 12)
- 6:   Asexual crossover process (see Algorithm 14)
- 7:   Settlement of new corals (see Algorithm 15)
- 8:   Predation process (see Algorithm 16)
- 9:   Evaluate the new population in coral reef
- 10: **end for**
- 11: Show the best individual of the coral reef

**5.3.1 Proceso de comienzo del algoritmo**

El algoritmo comienza con la generación de una estructura bidimensional vacía de tamaño  $N \times M$  que representa el arrecife o hábitat para los futuros corales. El arrecife completa  $N_0$  de sus celdas con corales aleatorios. Cada uno representa una solución al problema MNDP compuesta por un conjunto variable de estaciones base cuyos tipos pertenecen al conjunto  $\mathcal{T}$  y ubicadas en localizaciones de  $\mathcal{P}$ . Además de ser de creación aleatoria, su disposición en el arrecife es también al azar. La cantidad  $N_0$  se determina como el producto de la capacidad total del arrecife  $N_{ind}$  y del factor de ocupación del arrecife  $\rho$ , como muestra la Ecuación 5.2.

$$N_{ind} = N \cdot M \quad (5.1)$$

$$N_0 = N_{ind} \cdot \rho \quad (5.2)$$

El número de estaciones base para un individuo de coral de nueva creación siempre se encuentra en el rango  $[N_{MinB}, N_{MaxB}]$ . Este intervalo se calcula como de la misma forma que en el EA, expuesto en el Capítulo 4. Una vez compuesto el conjunto inicial de  $N_0$  corales, éstos son evaluados calculando su valor de *fitness*. El Algoritmo 10 recoge todas estas las acciones llevadas a cabo en el proceso de inicio del algoritmo.

**5.3.2 Proceso de eliminación de réplicas**

En la naturaleza resulta muy beneficioso que el arrecife crezca de forma rápida aun a costa de réplicas exactas de algunos corales. En el algoritmo CRO, la reproducción asexual otorga a los mejores individuos de cada colonia mayores probabilidades de replicarse frente al resto de la población. Sin embargo, hemos comprobado la tendencia que tiene el coral con mejor valor de *fitness* a extenderse de forma dominante en generaciones avanzadas. Es necesario controlar el número de réplicas para permitir suficiente diversidad de la población, a pesar del coste computacional que esta operación conlleva.

Tras numerosas simulaciones del algoritmo CRO hemos llegado a un nivel de compromiso: el número de réplicas que se mantiene en la población es la mitad de las encontradas siempre y cuando existan más de tres individuos idénticos. Para aliviar el coste computacional que conlleva

---

**Algorithm 10** Pseudo-código del proceso de generación de la colonia inicial en el algoritmo CRO

---

**Require:** Value for  $N_0$  (initial number of individuals) and range  $[N_{MaxB}, N_{MinB}]$  for number of BTSs of each individual

**Ensure:** Feasible population of coral reef

- 1: **for** each individual of  $N_0$  **do**
  - 2:   Select the number of BTSs  $num_{BTS}$  belongs to range  $[N_{MaxB}, N_{MinB}]$
  - 3:   Select  $num_{BTS}$  random and different locations from  $\mathcal{P}$
  - 4:   Select  $num_{BTS}$  random types from  $\mathcal{T}$
  - 5:   Ordering locations in ascending order
  - 6:   Evaluate the individual
  - 7:   Placement in the reef
  - 8: **end for**
- 

el proceso de eliminación solo se aplica sobre el mejor individuo si no se ha encontrado otro mejor durante al menos en 5 generaciones consecutivas. El Algoritmo 11 muestra el pseudo-código de detección y eliminación de réplicas del individuo con mejor valor de *fitness*.

---

**Algorithm 11** Pseudo-código del proceso de eliminación de réplicas en el algoritmo CRO

---

**Require:** Coral reef population with uncontrolled number of copies

**Ensure:** Feasible coral reef population with controlled number of copies

- 1: Select the best individual  $n_1$  of the coral reef
  - 2: Find the set  $\mathcal{N}_{n_1}$  of individuals with the same *fitness* value as  $n_1$
  - 3: **if** dimension  $D$  of  $\mathcal{N}_{n_1} > 3$  **then**
  - 4:   **while**  $D > \lceil D/2 \rceil$  **do**
  - 5:     Select an individual  $n_1^i$  in  $\mathcal{N}_{n_1}$
  - 6:     **if**  $n_1^i$  and  $n_1$  have same number of BTS **then**
  - 7:       **if**  $n_1^i$  and  $n_1$  have same locations for each BTS **then**
  - 8:         Remove individual  $n_1^i$  of coral reef
  - 9:       **end if**
  - 10:    **end if**
  - 11:   **end while**
  - 12: **end if**
- 

### 5.3.3 Métodos de reproducción de corales

En esta sección vamos a desarrollar los tres métodos de reproducción de individuos implementados en el algoritmo CRO inspirados en los tipos de reproducción de corales en la naturaleza. Tal y como presentamos en la Sección 5.2.3, los métodos de reproducción en el algoritmo CRO son:

- Reproducción sexual externa: se basa en el cruce tradicional de dos corales que se recombinan de forma ordenada para generar un único descendiente. Los individuos descendientes contienen información de ambos progenitores en igual o distinta medida.
- Reproducción sexual interna: tiene cierta semejanza a la mutación del EA. Los individuos sometidos a esta operación son duplicados y a su vez, su copia es mutada con una determinada probabilidad.
- Reproducción asexual: está inspirado en la reproducción por división de corales en la naturaleza. En este caso interviene un único progenitor y consiste en generar una copia

exacta al él mismo sobre la que pueden tener lugar o no pequeños cambios dirigidos a mejorar su valor de *fitness*. En el algoritmo CRO se permite este tipo de reproducción de una forma muy controlada, otorgando a éstos mayores posibilidades de sobrevivir y extenderse en el arrecife. El control se realiza mediante el proceso de eliminación de réplicas explicado en la Sección 5.3.2.

La tasa de reproducción en la naturaleza es muy elevada debido a las escasas posibilidades de supervivencia de nuevos corales. En el algoritmo CRO se permite que todos los individuos se reproduzcan por alguno de los métodos expuestos, la mayoría por el método de reproducción sexual externa y una minoría según las técnicas sexual interna (clonación seguido de mutación) y reproducción asexual (clonación de uno de los mejores especímenes del arrecife). Para ello, agrupamos los individuos en dos conjuntos de distinto tamaño. Para la reproducción sexual externa se recogen aleatoriamente el  $Pb_{ext}\%$  de los corales, para la reproducción sexual interna, el  $Pb_{int}\%$  de la población. De forma añadida se realiza la clonación (reproducción asexual) de un único individuo siempre y cuando se encuentre entre los  $p_{best}\%$  mejores. Esta réplica puede sufrir alguna alteración en el proceso.

### 5.3.3.1 Método de reproducción sexual

Como hemos comentado arriba, reunimos los métodos de reproducción sexual interna y externa, asignando probabilidades  $Pb_{ext}$  y  $Pb_{int}$  respectivamente. Lo esperable es que cada uno de los individuos tenga un tipo u otro de reproducción sexual (externa o interna), es decir, que la suma de ambas probabilidades sea la unidad ( $Pb_{ext} + Pb_{int} = 1$ ). En caso de superar la unidad, habrá uno o más corales que tengan más oportunidades de reproducción. En caso de ser la suma anterior inferior a la unidad implicará que algún individuo no tenga opción a reproducirse de esta forma. El Algoritmo 12 muestra el pseudo-código detallado del agrupamiento de individuos aleatorios en dos conjuntos diferentes  $E$  para la operación externa e  $I$  para la sexual interna.

---

#### Algorithm 12 Pseudo-código del proceso de reproducción sexual en el algoritmo CRO

---

**Require:** Coral reef with  $\mathcal{C}$  individuals,  $Pb_{ext}$  probability of external reproduction,  $Pb_{int}$  probability of internal reproduction

**Ensure:** A pool of new feasible individuals

- 1: Randomize population
  - 2: Select set  $E = \{\mathcal{C} \times Pb_{ext}\}$ , individuals for external crossover
  - 3: Pair up set  $E$
  - 4: **for** each couple of  $E$  **do**
  - 5:   Generate a new individual through TBC or DMC crossover
  - 6:   Add the new individual to the pool
  - 7: **end for**
  - 8: Set  $I = \{\mathcal{C} \times Pb_{int}\}$ , individuals for internal crossover (if  $Pb_{ext} + Pb_{int} = 1$  then  $E \cap I = \emptyset$ )
  - 9: **for** each individual of  $I$  **do**
  - 10:   Duplicate and mutate the individual
  - 11:   Add the new individual to the pool
  - 12: **end for**
- 

El método de reproducción sexual externa coincide con el método de cruce del EA desarrollado en el Capítulo 4, donde dos individuos se recombinan para generar un nuevo descendiente. En el algoritmo CRO también utilizamos los operadores de cruce definidos para el EA: el TBC (Template-Based Crossover) y el DMC (Direct Merge Crossover). El operador TBC consiste en la aplicación de una máscara binaria aleatoria para determinar qué localizaciones y tipos de

estaciones base de uno u otro progenitor son seleccionados para componer el individuo descendiente. La variante DMC consiste en crear un nuevo coral que engloba todas las estaciones base de ambos progenitores y confiar en un proceso de búsqueda local que elimine las estaciones que considere prescindibles. El pseudo-código para ambas variantes puede revisarse en el Capítulo 4 con los títulos: “Pseudo-código del operador de cruce TBC” y “Pseudo-código del operador de cruce DMC”.

Como ya se ha explicado, la reproducción sexual interna consiste técnicamente en una réplica junto con una mutación. En la práctica implementamos una mutación como un cruce de tipo TBC con un individuo aleatorio con el mismo número de estaciones base.

Durante la etapa final de desarrollo de este trabajo tratamos de incluir variaciones en algunas de las partes del algoritmo. Concretamente modificamos la mutación aplicada en la reproducción sexual interna. El Algoritmo 13 muestra el pseudo-código del proceso de mutación. Pueden observarse tres variantes identificadas con los nombres de *traditional*, *total random* y *no mutation*. En el primero la máscara binaria es aleatoria y por lo tanto la probabilidad de cambio en cada elemento del individuo es del 50%. En el caso de *total random* el individuo se sustituye completamente por el coral generado aleatoriamente. El último caso (*no mutation*), la probabilidad de mutación es nula y por lo tanto se mantiene una réplica exacta del progenitor. La operación finaliza realizando una búsqueda local si ésta es requerida, ya que debido a la carga computacional que supone, no debemos aplicarla de forma indiscriminada.

---

**Algorithm 13** Pseudo-código para el operador de mutación en el algoritmo CRO

---

**Require:** An individual of length  $L$ , type of mutation  $t_m$

**Ensure:** Feasible mutated individual

- 1: **switch** (type of mutation  $t_m$ )
  - 2: **case** traditional:
  - 3:   Generate a random binary mask with length  $L$
  - 4: **case** total random:
  - 5:   Generate a ones vector with length  $L$
  - 6: **case** no mutation:
  - 7:   Generate a zeros vector with length  $L$
  - 8: **end switch**
  - 9: In locations part  $p_i$ , change locations to other random ones where mask has bit '1'
  - 10: In typing part  $t_i$ , change types to other random ones where mask has bit '1'
  - 11: Ordering stations
  - 12: Local searching
- 

### 5.3.3.2 Método de reproducción asexual

Este método ofrece la posibilidad de replicar a uno o varios individuos y realizar pequeños cambios que mejoren su valor de *fitness*. Para evitar una posible falta de diversidad en la población, la operación se aplica únicamente sobre los mejores individuos de la misma. En cada generación se selecciona un subconjunto de la población de los individuos con mejor valor de *fitness* y definido por el porcentaje (% $p$ ). La réplica se realiza entonces en uno y solo uno de los integrantes de este conjunto elegido de forma aleatoria. A continuación se muestra el pseudo-código en el Algoritmo 14. Existe una infinidad de cambios estratégicos (*strategic local searching*) que pueden mejorar el valor de la función de evaluación de un coral. Los cambios que aplicamos en este trabajo se centran en modificar el tipo de alguna estación base o bien actuar sobre la estación base que da servicio al menor número de localizaciones de la solución. Esta estación puede ser eliminada por completo del coral (con un 33% de posibilidades), cambiar su tipo por otro al azar con un

33% de probabilidad, o bien cambiar el tipo de una estación base aleatoria del individuo. Hemos comprobado experimentalmente que este tipo de cambios resultan eficaces en etapas avanzadas del algoritmo.

---

**Algorithm 14** Pseudo-código del proceso de reproducción asexual en el algoritmo CRO
 

---

**Require:** Coral reef with  $\mathcal{C}$  individuals, percentage  $\%p$  of the best individuals to have asexual crossover

**Ensure:** A pool with an unique individual

- 1: Set  $A = \{ \mathcal{C} \times \%p \text{ of the best corals in } \mathcal{C} \}$
  - 2: Select a random individual of  $A$
  - 3: Copy of this individual to the pool
  - 4: Apply strategic local searching to the copy
  - 5: Add the new individual to the pool
- 

### 5.3.4 Método de establecimiento en el arrecife

Los corales generados mediante cualquiera de los anteriores métodos de reproducción tratan de establecerse en la estructura del arrecife y alcanzar de esta manera la siguiente generación. Una cualidad propia del algoritmo CRO es que los corales o plánulas de nueva creación son almacenadas en una memoria o *pool*. Tras finalizar el proceso reproductivo tiene lugar la operación de asentamiento para todos ellos. Pretendemos modelar de esta forma lo que ocurre en la naturaleza, donde las plánulas flotan en el agua durante días alrededor del coral y tras alcanzar el nivel de desarrollo suficiente de tentáculos y boca, se desprenden del *cilio* que les permite nadar y se adhieren al arrecife. Si los corales fueran asentándose según van siendo creados, los últimos se encontrarían en desventaja frente a los que primero aparecieron. Retenidos todos en el *pool* deben esperar el momento adecuado para comenzar el asentamiento.

El nuevo coral tiene un número máximo de intentos u oportunidades  $\mathcal{N}_{Ch}$  para lograr establecerse en un hueco sobre la estructura de arrecife. En cada intento selecciona una celda al azar. Ésta puede estar vacía o bien ocupada por un anterior coral residente. Si la celda se encuentra libre, el nuevo individuo procede simplemente a establecerse como propietario. En caso de estar el espacio previamente ocupado, tiene lugar una competición entre los dos contrincantes. El vencedor será aquel con mejor valor de *fitness*, convirtiéndose en el propietario del lugar hasta otra posible contienda. Si la victoria corresponde a la nueva plánula, el anterior ocupante muere sin más. En caso contrario el nuevo individuo continúa la búsqueda hasta completar  $\mathcal{N}_{Ch}$  intentos. Si después de  $\mathcal{N}_{Ch}$  oportunidades la plánula no logra el asentamiento, es eliminada de la población. El Algoritmo 15 muestra la secuencia ordenada de las acciones que se llevan a cabo en la operación de establecimiento en el arrecife. El orden en el que los nuevos corales realizan el asentamiento es aleatorio para ofrecer a todos las mismas oportunidades.

### 5.3.5 Proceso de depredación en el algoritmo

Tras el asentamiento de las nuevas plánulas en el arrecife, tiene lugar una etapa de depredación o aniquilamiento de algunos individuos. Este hecho modela el efecto devastador que sobre el coral ejercen ciertos agentes externos debido a la intervención humana (como choques de elementos sólidos contra el arrecife) o a factores naturales (como son las tormentas y fuertes corrientes). El proceso destruye algunos corales con escasa adaptación al medio (menor adherencia a la estructura de coral) y otros de manera fortuita. En el algoritmo CRO seleccionamos un grupo de tamaño variable que identifica los individuos con peor valor de *fitness*. Dicho grupo queda



---

**Algorithm 15** Pseudo-código para el proceso de asentamiento en el arrecife en el algoritmo CRO

---

**Require:** Coral reef, pool of new individuals to settle,  $N_{Ch}$  as maximum number of chances to settlement

**Ensure:** A different coral reef

```

1: for each random individual of the pool do
2:    $chance = 0$ 
3:   while  $chance < N_{Ch}$  do
4:      $chance + 1$ 
5:     Select a random space in coral reef
6:     if position is empty then
7:       Individual settles in the space
8:     else
9:       if fitness of the individual better than fitness of the owner then
10:        Individual settles in the space
11:      end if
12:    end if
13:  end while
14: end for

```

---

definido por un porcentaje  $\%p$ . Cada uno de los integrantes de este grupo serán eliminados de la población con una probabilidad  $Pb_{pred}$ . Asignando los valores adecuados a ambos parámetros podemos reforzar el efecto destructor sobre los individuos peor adaptados, con un porcentaje  $\%p$  reducido y una probabilidad  $Pb_{pred}$  alta, o sobre el coral en general debido a la intervención de un agente externo humano o no humano, con un porcentaje  $\%p$  elevado y una reducida probabilidad  $Pb_{pred}$ . Estos valores pueden ser modificados según avanzan las generaciones durante la ejecución. Finalmente mostramos el pseudo-código del proceso de depredación en el Algoritmo 16.

---

**Algorithm 16** Pseudo-código del proceso de depredación en el algoritmo CRO

---

**Require:** Coral reef with  $\mathcal{C}$  individuals, percentage  $\%p$  of individuals to be preied and predation probability  $Pb_{pred}$

**Ensure:** Coral reef with less or equal number of individuals

```

1: Set  $A = \{\mathcal{C} \times \%p \text{ of the worst corals in } \mathcal{C}\}$ 
2: for each individual of  $A$  do
3:   Delete the individual with  $Pb_{pred}$  probability
4: end for

```

---



## Capítulo 6

# Resultados

### 6.1 Especificaciones del problema MNDP en un caso real

Los algoritmos implementados en este trabajo han sido ejecutados basándose en mediciones reales de campo electromagnético en la ciudad de Alcalá de Henares durante el año 2006. Estos datos fueron recogidos durante la realización del proyecto de investigación mencionado en el Capítulo 1: “Estudio de las emisiones radioeléctricas en las bandas de frecuencia asignadas a los servicios de telefonía móvil, en la ciudad de Alcalá de Henares”.

En el presente capítulo describimos los resultados obtenidos mediante un conjunto de simulaciones realizadas en cada uno de los cinco algoritmos: *Greedy*, PSO, HS, EA y CRO. Los tres primeros algoritmos se emplean fundamentalmente para comparar la eficiencia y eficacia de los dos últimos, que constituyen la aportación principal del trabajo. Todos los algoritmos han sido ejecutados sobre un procesador 1,6 GHz Intel Corel 2 Duo con memoria de 2 GHz y 667 MHz. Los resultados de estos algoritmos solo son comparables si todos ellos parten de una base común de parámetros generales. A continuación enumeramos el conjunto de parámetros generales y abordamos cada uno de forma individual.

- Área de estudio  $A$ .
- Conjuntos de puntos  $\mathcal{O}$ ,  $\mathcal{P}$  y  $\mathcal{P}'$ . Estos conjuntos los consideramos idénticos en esta parte del trabajo. En el Anexo B se ofrece un ejemplo considerando estos tres conjuntos disjuntos.
- Conjunto de diferentes tipos de estaciones base  $\mathcal{T}$ , particularizando para una determinada frecuencia de trabajo de las estaciones base: 900 MHz, 1800 MHz o 2100 MHz.
- Función de evaluación.
- Número de individuos en cada generación.
- Rango de número de estaciones base para la generación de nuevos individuos aleatorios.
- Número de evaluaciones realizadas en cada simulación.

El área  $A$  a estudiado comprende  $40,281745 \text{ Km}^2$  encerrado en un rectángulo aproximadamente de  $9 \times 7 \text{ Km}$ . Dentro de esta superficie  $A$  se establece el conjunto de 509 puntos de medida  $\mathcal{O} = \{\mathbf{o}_i\}$ ,  $i \in \{1, \dots, O\}$ . Cada punto  $\mathbf{o}_i$  queda definido por sus coordenadas  $(x_i^o, y_i^o)$  y por su altitud en metros. Además se cuenta con un registro de medidas experimentales de campo eléctrico (V/m) para cada elemento del conjunto  $\mathcal{O}$ . La altura de cada punto de  $\mathcal{O}$  es un dato necesario para realizar el cálculo del campo eléctrico sobre dicho punto debido a un

conjunto de estaciones base trabajando en  $A$ . La Tabla 6.1 recoge el tramo inicial del conjunto  $\mathcal{O}$  utilizado. En esta parte del trabajo consideramos idénticos los conjuntos  $\mathcal{O}$ ,  $\mathcal{P}$  y  $\mathcal{P}'$ , siendo  $\mathcal{P} = \{\mathbf{p}_i\}$ ,  $i \in \{1, \dots, P\}$  el conjunto de puntos factibles para la instalación de los equipos que integran las estaciones base y  $\mathcal{P}' = \{\mathbf{p}'_i\}$ ,  $i \in \{1, \dots, M\}$  el conjunto de puntos de interés para ofrecer el servicio de telecomunicación.

**Tabla 6.1:** Extracto del conjunto  $\mathcal{P}$  utilizado en las simulaciones, formado por 509 puntos pertenecientes a la ciudad de Alcalá de Henares.

Identificador	Coordenada $x$	Coordenada $y$	Altura (m)	E(V/m)
1	468607	4481266	15	0,24
2	468514	4481395	26	0,13
3	468850	4481216	36	0,25
4	468720	4481175	18	0,06
5	468837	4481432	30	0,11
...	...	...	...	...
509	469488	4483782	26	0,19

El conjunto de estaciones base  $\mathcal{T}$  empleado en las simulaciones consta de tres tipos diferentes trabajando a una frecuencia de 1800 MHz, siendo  $\mathcal{T} = \{t_i\}$ ,  $i \in \{1, \dots, T\}$ . Obviamente, este conjunto puede extenderse a otros tipos de antenas para las frecuencias de 900 MHz y 2100 MHz en redes GSM. Cada tipo de estación base queda definido por su potencia de transmisión (en dBw) y su coste ( $\cdot 10^3\text{€}$ ). La Tabla 6.2 recoge los valores de potencia de transmisión y coste de los tipos de antenas utilizados para las simulaciones presentadas.

**Tabla 6.2:** Tipos de estaciones base empleados para las simulaciones para una frecuencia de trabajo de 1800 MHz.

Id. tipo	Pot. transmisión (dBw)	Coste ( $\cdot 10^3\text{€}$ )
A	6	100
B	10	200
C	14	300

En lo que respecta a la función de evaluación o función de *fitness*, es imprescindible utilizar la misma función de evaluación en todos los algoritmos a comparar. Definimos esta función en el Capítulo 4 según la Ecuación 6.1, donde  $\Delta E_T$  es el incremento total de campo eléctrico medido en V/m y debido al conjunto de estaciones base de la nueva red,  $\mathcal{C}_{\mathcal{T}}$  refleja el coste total medido en miles de euros ( $10^3\text{€}$ ) del despliegue de dichas estaciones base y  $P_{no\ cober}$  representa el porcentaje de puntos sin servicio sobre el conjunto de puntos objetivo  $\mathcal{P}'$ . Los diferentes algoritmos desarrollados tratan de optimizar esta función encontrar valores reducidos para cada una las tres variables de forma conjunta.

Las constantes  $k_1$ ,  $k_2$  y  $k_3$  son los coeficientes que permiten la adicción compuesta de magnitudes diferentes, de modo que el producto de cada coeficiente por la magnitud correspondiente resulte un valor adimensional. El valor de estos coeficientes debe ser el mismo en todos los algoritmos si se desea comparar los valores de *fitness* proporcionados por cada algoritmo. Es importante asignar valores idóneos a estos coeficientes puesto que son los responsables dar mayor o menor peso a cada una de las tres magnitudes protagonistas de la función de *fitness*. Debemos modificar el valor del coeficiente correspondiente hacia arriba o hacia abajo según se desee potenciar su efecto o suavizarlo frente a las otras dos variables. En este trabajo se orienta a

encontrar soluciones con reducida exposición de radiación electromagnética, de modo que los valores numéricos de los tres coeficientes son elegidos conjuntamente teniendo en cuenta este interés.

$$\mathcal{F} = k_1 \cdot \Delta E_T + k_2 \cdot \mathcal{C}_T + k_3 \cdot P_{no\ cober}(\%) \quad (6.1)$$

Los cálculos de cada magnitud se desarrollaron en el Capítulo 3. Aquí solo exponemos los valores numéricos utilizados en las ecuaciones allí planteadas para el cálculo de la contribución al campo eléctrico y el cálculo del porcentaje de cobertura/no cobertura de la red. En el cálculo del incremento de campo eléctrico  $\Delta E_T$  y considerando una frecuencia de trabajo de 1800 MHz empleamos el modelo de *Cost231-Hata*, válido en el rango  $1500 \leq f \leq 2000$  MHz. En este modelo hemos fijado los valores de el factor de corrección  $C_m$  y la altura del terminal receptor  $h_{Mobile}$  (por ejemplo el teléfono móvil del usuario) de la siguiente manera:

$$\begin{aligned} C_m &= 3 \text{ dB en áreas metropolitanas} \\ h_{Mobile} &= 1,75 \text{ m} \end{aligned}$$

Para cuantificar el porcentaje de no cobertura de la red sobre  $A$  se calcula el radio teórico de cobertura de cada estación base y se registran aquellos puntos de  $\mathcal{P}'$  que quedan fuera cobertura de todas las estaciones base de la red. A partir de la Ecuación 6.2 y con una sensibilidad del receptor a  $S_r = -94$  dBm, se determina el valor de las pérdidas de propagación  $L_p$ . Finalmente mediante las ecuaciones proporcionadas en el Capítulo 3 de los modelos de *Okumura-Hata* y *Cost231-Hata*, se obtiene el radio de cobertura como valor máximo de la distancia  $d$  (Km) en las Ecuaciones 3.10 y 3.12 respectivamente.

$$L_p(\text{dB}) = P_T(\text{dBW}) - S_r(\text{dBm}) + 30 \quad (6.2)$$

Para establecer los valores numéricos de los coeficientes  $k_1$ ,  $k_2$  y  $k_3$  de la función de evaluación, se realizaron diversas simulaciones con el algoritmo evolutivo variando estos coeficientes. Las soluciones extraídas fueron analizadas respecto a cada una de las tres magnitudes de la función: incremento de campo eléctrico, coste y porcentaje de no cobertura de la red. El número de estaciones base también puede ser un parámetro a considerar, aunque indirectamente queda medido en el coste del despliegue de red. Las Tablas 6.3 y 6.4 muestran los resultados obtenidos con tres grupos de valores diferentes para  $k_1$ ,  $k_2$  y  $k_3$ . Experimentalmente comprobamos que variaciones en el coeficiente  $k_3$  no afectaban tanto a la variación de calidad en las soluciones del algoritmo evolutivo como las alteraciones en los otros dos coeficientes ( $k_1$  y  $k_2$ ). Por esta razón y una vez establecido un valor de  $k_3$  adecuado para el rango de los otros dos, se mantuvo constante en  $k_3 = 1,75$ . Con cada combinación de coeficientes se realizaron 60 simulaciones en poblaciones de 50 individuos. Los valores promedio se recogen en la Tabla 6.3 y los mejores individuos o soluciones encontradas para cada grupo figuran en la Tabla 6.4. En ambas tablas hemos destacado en color rojo la segunda combinación de coeficientes:  $k_1 = 15 (V/m)^{-1}$  y  $k_2 = 5 \cdot 10^{-5} (\cdot 10^3 \text{€})^{-1}$ . Esta combinación ofrece un buen nivel de compromiso con respecto a las otras dos, en el sentido que el segundo grupo encuentra soluciones no tan costosas como el primero (unos  $10^6$  € por debajo calculados estos costes a partir de la Tabla 6.2) y con una mejora apreciable sobre la tercera combinación en lo que al grado de cobertura de la red se refiere (la combinación segunda tiene un porcentaje de no cobertura que es menos de la mitad del obtenido en la tercera), estando los valores de incremento de campo eléctrico de ambas bastante próximos (3,2 mV/m y 3,1 mV/m entre las mejores soluciones en la Tabla 6.4). Con estos datos se determinó que para todos los algoritmos se utilizarían los valores recogidos en la Tabla 6.5.

**Tabla 6.3:** Resultados promedios obtenidos a partir de 60 simulaciones con el algoritmo evolutivo con diferentes valores de los coeficientes  $k_1$  y  $k_2$  en la función de evaluación  $\mathcal{F}$ . El coeficiente  $k_3$  se ha mantenido constante  $k_3 = 1,75$ .

		Valores promedio (60 simulaciones)			
$k_1(V/m)^{-1}$	$k_2(\cdot 10^3\text{€})^{-1}$	Núm. estaciones	$\Delta E(mV/m)$	$\mathcal{C}(\cdot 10^6\text{€})$	(%) No cobertura
10	$1 \times 10^{-5}$	14,4	4,48	3,8883	0,28
15	$5 \times 10^{-5}$	11,417	4,4	2,8383	1,94
15	$1 \times 10^{-4}$	9,383	4,41	2,2167	4,53

**Tabla 6.4:** Mejores resultados obtenidos a partir de 60 simulaciones del algoritmo evolutivo con diferentes valores de los coeficientes  $k_1$  y  $k_2$  en la función de evaluación  $\mathcal{F}$ . El coeficiente  $k_3$  se ha mantenido constante  $k_3 = 1,75$ .

		Valores mínimos de 60 simulaciones			
$k_1(V/m)^{-1}$	$k_2(\cdot 10^3\text{€})^{-1}$	Núm. estaciones	$\Delta E(mV/m)$	$\mathcal{C}(\cdot 10^6\text{€})$	(%) No cobertura
10	$1 \times 10^{-5}$	12	3,545	3,3	0
15	$5 \times 10^{-5}$	9	3,200	2,3	0,59
15	$1 \times 10^{-4}$	8	3,100	1,9	2,75

**Tabla 6.5:** Valores finales utilizados para los coeficientes  $k_1$ ,  $k_2$  y  $k_3$  de la función de evaluación  $\mathcal{F}$  para los cinco algoritmos implementados.

Coficiente de $\mathcal{F}$	Valor	Unidades
$k_1$	15	$(V/m)^{-1}$
$k_2$	$5 \times 10^{-5}$	$(\cdot 10^3\text{€})^{-1}$
$k_3$	1,75	-

Los algoritmos basados en poblaciones analizados (a excepción del algoritmo *Greedy* que trabaja con un único individuo en cada iteración) utilizan el mismo tamaño de población  $N_{ind} = 50$  individuos. En la primera iteración, cada individuo  $i$  es generado de manera aleatoria con un número de estaciones base  $n_i$  al azar. De este modo, cada una de sus estaciones base será de tipo aleatorio dentro del conjunto  $\mathcal{T}$  (Tabla 6.2) y estará ubicada en algún punto del conjunto de  $\mathcal{P}$  de la Tabla 6.1 seleccionado también de forma aleatoria. Para determinar  $n_i$  se establece un intervalo de valores  $[N_{MinB}, N_{MaxB}]$  adecuados según la extensión del área  $A$  a cubrir y a las características de potencia de transmisión las estaciones base disponibles. En el Capítulo 4 se detalló el método para establecer dicho intervalo: considerando el tipo de estación base con menor potencia de transmisión y el punto de  $\mathcal{P}$  con menor altura en el conjunto  $\mathcal{P}$ , se determina la superficie en  $m^2$  a la que puede ofrecer cobertura una estación base (suponiendo cobertura teórica circular). Este sería el peor de los casos dado que intensifica el número mínimo de estaciones de este tipo para cubrir  $A$ . Finalmente se extiende este resultado a la superficie total  $A$  (para mayor detalle consultar el Capítulo 4). Para la determinación del intervalo  $[N_{MinB}, N_{MaxB}]$  hemos considerado los siguientes valores:

- Área  $A = 40,281745 \text{ Km}^2$ ;
- Mínima potencia de las antenas en  $\mathcal{T}$ :  $P_T = 6 \text{ dBW}$
- Frecuencia de trabajo de la antena: 1800 MHz

- Altura mínima de entre las posibles localizaciones en  $\mathcal{P}$ :  $h_{BTS} = 15 \text{ m}$
- Radio de cobertura calculado mediante la ecuación que describe el modelo de *Cost231-Hata* en el Capítulo 3:
 
$$r_{cob} = 457,3673 \text{ m}$$

$$B_{max} = \lceil \frac{A}{\pi r_{cob}^2} \rceil = 62 \text{ BTSs}$$

$$N_{MaxB} = \lceil 0,4B_{max} \rceil = 25 \text{ BTSs}$$

$$N_{MinB} = \lceil 0,3MaxB \rceil = 9 \text{ BTSs}$$

Durante el resto de iteraciones será el propio algoritmo el responsable de modificar el número de estaciones base de los individuos. Los algoritmos implementados tienen la capacidad de reducir y/o aumentar el número de estaciones de un individuo. De modo que una estimación errónea de  $n_i$  durante la primera iteración no conduce al fracaso del algoritmo, solo retrasa la convergencia del mismo.

Finalmente consideramos el último parámetro enumerado en la página 93, siendo éste el número total de evaluaciones  $N_E$  de la función de *fitness* en cada simulación. Para determinar la duración de una simulación con cualquiera de los cinco algoritmos es necesario establecer un número de evaluaciones  $N_E$  sobre las soluciones propuestas por cada algoritmo en cuestión. Para que los resultados de los diferentes algoritmos sean comparables se igualan el número de evaluaciones  $N_E$  entre ellos. En los apartados siguientes se explica el cálculo de  $N_E$  en cada caso.

## 6.2 Especificaciones para el EA

En este caso el número de evaluaciones de la función de *fitness* en la simulación  $i$  se puede estimar de forma simple como  $N_E = \mathcal{N}_{ind} \times G_i$ , donde  $\mathcal{N}_{ind}$  es el número de individuos en la población y  $G_i$  es el número de generaciones que alcanza el algoritmo para la simulación  $i$ . Tal y como se comenta en el apartado de condiciones de parada para el algoritmo evolutivo propuesto en el Capítulo 4, el algoritmo finaliza cuando se cumple al menos una de las dos condiciones de parada siguientes: se produce la convergencia de la población en una generación o bien el algoritmo alcanza un máximo de generaciones  $G_{Max}$ . Decimos que la población converge en una generación cuando se cumple la Ecuación 6.3 donde el valor de *fitness* del mejor individuo  $\mathcal{F}_{best}$  es muy similar al valor medio de la función de evaluación  $\bar{\mathcal{F}}$  para toda la población de la dicha generación, es decir:

$$\bar{\mathcal{F}} - \epsilon \leq \mathcal{F}_{best} \leq \bar{\mathcal{F}} + \epsilon \quad (6.3)$$

Por otro lado,  $G_{Max}$  se puede considerar una cota inferior para el número de generaciones que hace converger a la población en el sentido que ejecutar el algoritmo durante un número mayor de generaciones no garantiza mejores resultados si es probable que se haya alcanzado la convergencia de la población. Experimentalmente se ha obtenido que para una población de  $\mathcal{N}_{ind} = 50$  individuos y con un valor de  $\epsilon$  de 1%, la población converge en media tras 196 generaciones. Con un valor de  $\mathcal{N}_{ind} = 50$  individuos y  $G_i = 196$  generaciones se obtiene un número total de evaluaciones de la función de *fitness* de  $N_E = 9800$  evaluaciones. Este valor será una cota de referencia para el resto de algoritmos. En todos ellos y siempre que haya sido posible, el número de generaciones o iteraciones se ha redondeado a 200. A continuación se detallan las características específicas en la ejecución del algoritmo evolutivo.

- Durante el desarrollo de este trabajo se han realizado en torno a unas 150 simulaciones del EA. Con el objetivo de comparar los diferentes algoritmos, hemos elegido las últimas 60 simulaciones consecutivas, que componen los resultados que se presentan en este documento.
- Proceso de selección de individuos. Éste tiene lugar en dos instantes diferentes en cada generación  $G_i$ : durante la formación de parejas progenitores antes de la operación de cruce y en la selección de individuos supervivientes para la generación  $G_{i+1}$ . Las parejas de progenitores se forman al azar, sin repetición y de manera equiprobable. De esta forma y puesto que el tamaño de la población es  $N_{ind} = 50$  individuos, son distribuidos en  $N_{ind}/2 = 25$  parejas participando cada individuo en una única pareja. Anteriormente a este tipo de selección, hemos probado con el método de la ruleta explicado en el Capítulo 4 permitiendo, en principio, mayor participación a individuos con mejor valor de *fitness*, sin embargo no obtuvimos mejores resultados. Finalmente optamos por la equiprobabilidad en la población persiguiendo mayor diversidad, teniendo en cuenta que el número de descendientes no es muy elevado.

La selección de individuos que alcanzan la siguiente generación se realiza mediante torneo. Entre la población total compuesta por progenitores y descendientes se realizan  $N_{ind}$  torneos de dos participantes sin repetición. En caso de no ser posible obtener  $N_{ind}$  parejas sin repetición, la población superviviente se completa de forma elitista con los mejores individuos. Se puede optar por torneos permitiendo la repetición de individuos en tantos torneos necesarios hasta completar los  $N_{ind}$  supervivientes destinados a la generación siguiente. No consideramos que los resultados finales del algoritmo dependan fuertemente de la condición de repetición puesto que ésta afecta a un porcentaje reducido de individuos. Los resultados aquí recogidos se han obtenido con torneos sin repetición: con el valor  $N_{ind} = 50$  individuos progenitores utilizado en las simulaciones y un descendiente por pareja se obtiene una población previa al torneo de 75 individuos disponibles. Dado que solo es posible realizar  $\lfloor 75/2 \rfloor = 37$  torneos sin repetición, la población se completa con los  $50 - 37 = 13$  mejores individuos de la población disponible y no ganadores en ningún torneo.

- Operadores de cruce: como ya hemos indicado anteriormente, el cruce de dos individuos genera un único y nuevo descendiente a partir de sus progenitores. Los resultados presentados en este Capítulo son obtenidos a partir de uso combinado de los dos operadores detallados en el Capítulo 4: el operador TBS (Template-Based Crossover) y el operador DMC (Direct Merge Crossover). Tras simulaciones con diferentes porcentajes para ambos tipos de cruces, decidimos realizar un cruce de tipo TBS por cada tres de tipo DMC (25 % y 75 %, respectivamente). Damos mayor prioridad al cruce DMC porque permite que la función de búsqueda local decida qué estaciones quedan definitivas en el individuo en base al criterio de cobertura y no al azar inherente en la operación de cruce. Obviamente esta decisión conlleva mayor carga computacional, sin embargo es asumible para el tiempo total de ejecución de simulación (25 minutos en media en un procesador 1,6 GHz Intel Corel 2 Duo).
- El proceso de mutación altera elemento a elemento y de forma aleatoria las partes del individuo, tanto la ubicación de las estaciones base como la tipología de éstas. Habitualmente la probabilidad de mutación no supera el 5%. Tras varias pruebas preferimos aumentar esta probabilidad hasta el 55 % aún a costa de mayor carga computacional para ofrecer mayor diversidad. Pensamos que otra posibilidad sería haber optado por trabajar con una población más numerosa manteniendo los márgenes habituales de probabilidad de



mutación.

- Por último, el proceso de reparación del individuo empleado para la obtención de los resultados expuestos consiste en que las estaciones base identificadas como “prescindibles” son eliminadas o bien desplazadas a otra ubicación disponible del conjunto  $\mathcal{P}$  con un 50 % de probabilidad. Una estación base se califica como estación prescindible si cumple uno de los dos siguientes criterios: cubre menos del  $\beta = 0,5\%$  del total de puntos de interés o bien que ofrece servicio a localizaciones ya atendidas por otras estaciones base.

### 6.3 Especificaciones para el algoritmo Greedy

En general, con el algoritmo *Greedy* desarrollado no se obtienen resultados de alta calidad, aunque es el de menor complejidad computacional de todos los estudiados. Este algoritmo marca una cota inferior para la función de *fitness*. El algoritmo se ejecutó sucesivas veces computando el número de evaluaciones de la función de *fitness* para asegurar un número de evaluaciones total no inferior a  $N_E = 9800$ .

Para estimar el número de repeticiones o simulaciones necesario hasta alcanzar  $N_E \geq 9800$  evaluaciones, calculamos el número medio de evaluaciones que realiza el algoritmo por repetición. Según el pseudocódigo del algoritmo *Greedy* detallado en el Capítulo 4, el número de evaluaciones en cada repetición coincide con el número de iteraciones que a su vez es el número de estaciones base del individuo (cada iteración evalúa la conveniencia de eliminar una estación base). Este número de estaciones base del individuo es desconocido a priori, sin embargo sí se conoce el número medio de estaciones base de los individuos de nueva creación. El algoritmo genera un nuevo individuo y realiza una búsqueda local antes de comenzar el estudio iterativo. Durante la búsqueda local es esperable que se reduzca levemente el número de estaciones del individuo y por esta razón consideramos que el número medio de estaciones base en los individuos aleatorios representa una cota superior del número medio de evaluaciones por simulación.

Como explicamos en la Sección 6.1, los individuos de nueva creación toman un número de estaciones base aleatorio dentro del intervalo  $[N_{MinB}, N_{MaxB}] = [9 - 25]$ . El valor central de este intervalo es 17, y en media será el número de estaciones base asignadas a los individuos generados de esta forma. Con 17 evaluaciones por simulación y para alcanzar un número de evaluaciones  $N_E \geq 9800$  es necesario realizar del orden de  $N_E/17 = 576$  repeticiones del algoritmo *Greedy*. Dado que el valor de 17 es una cota superior del número medio de evaluaciones, el número de repeticiones así calculado ofrece una cota inferior. Experimentalmente el número de simulaciones necesarias para satisfacer  $N_E \geq 9800$  resultó ser 610 que incluían 9812 evaluaciones totales y en media 16 evaluaciones por repetición.

### 6.4 Especificaciones para el algoritmo PSO

De forma análoga al algoritmo evolutivo, hemos realizado una batería de 60 simulaciones aleatorias y consecutivas con el algoritmo PSO, estando cada simulación compuesta por 200 iteraciones. En cuanto al tamaño de la población, tal y como mencionamos en el Capítulo 4, la recomendación en [12, 97] es de poblaciones de 25 ó 30 individuos. Hemos realizado baterías de 60 simulaciones con poblaciones de 25 individuos bajo esta recomendación y también de 50 individuos para igualar las condiciones del resto de algoritmos. Esta discusión se lleva a cabo en las Tablas 6.6 y 6.7. La Tabla 6.6 ratifica la recomendación de [12, 97], ya que los resultados medios son bastante similares para ambos tamaños de población, donde poblaciones de unos 25 individuos alcanzan buen compromiso entre carga computacional y eficacia en resultados.

La Tabla 6.7 muestra los dos mejores individuos obtenidos en cada caso. Los resultados son algo mejores para las poblaciones de 50 individuos, lo que resulta coherente puesto que hay mayor diversidad en la población. El valor de *fitness* para el mejor individuo entre las poblaciones de 50 elementos es 0,28933, mientras que en poblaciones de 25 individuos, el mejor *fitness* se sitúa en 0,30464, esto significa un 5,29% más respecto al mejor *fitness* de la población más numerosa. La solución más óptima ofrece mejores valores de incremento de campo eléctrico (5 mV/m frente a 6,08 mV/m) y coste de la red ( $3,6 \cdot 10^6 \text{€}$  frente a  $4,2 \cdot 10^6 \text{€}$ ) con un porcentaje de cobertura superior al 98%.

A pesar de que las simulaciones con 50 individuos conllevan mayor carga computacional, vamos a considerarlas en la comparación con el resto de algoritmos desarrollados. Además coincidiendo el número de iteraciones (200 por simulación) y el tamaño de la población ( $N_{ind} = 50$  individuos), se igualan también el número de evaluaciones de la función de *fitness* a  $N_E = 9800$ .

**Tabla 6.6:** Valores promedio para los parámetros que caracterizan las soluciones encontradas con el algoritmo PSO para diferentes tamaños de población.

Tamaño de población	<i>fitness</i>	$\Delta E(\text{mV/m})$	$\mathcal{C}(\cdot 10^6 \text{€})$	% No cobertura	Núm. estaciones
25	0,3483	7,38	3,853	2,49	17,83
50	0,3387	7,00	3,793	2,52	17,83

**Tabla 6.7:** Mejores individuos obtenidos con el algoritmo PSO variando el tamaño de la población. (\*\*) Corresponden a las segundas mejores soluciones obtenidas en cada caso.

Tamaño de población	<i>fitness</i>	$\Delta E(\text{mV/m})$	$\mathcal{C}(\cdot 10^6 \text{€})$	% No cobertura	Núm. estaciones
25(**)	0,30733	6,78	4,0	1,18	17
25	0,30464	6,08	4,2	0,20	19
50(**)	0,30312	6,54	4,1	0,00	16
50	0,28933	5,00	3,6	1,96	18

Respecto a los parámetros específicos de este algoritmo, usamos modelo *inertia weight*, de modo que la velocidad de una partícula o solución en la iteración  $k+1$  se determina en función de su velocidad en la anterior iteración siguiendo la Ecuación 6.4:

$$v_i^{k+1} = \omega \cdot v_i^k + \varphi_1 \cdot rand_1 \cdot (pBest_i - x_i^k) + \varphi_2 \cdot rand_2 \cdot (g_i - x_i^k) \quad (6.4)$$

Con el fin de mejorar la eficiencia y tal y como se sugiere en [97], reducimos progresivamente el valor de *inertia weight* a lo largo de cada simulación, de manera exacta variamos linealmente su valor dentro del intervalo  $[0,9-0,1]$ . Con esta medida fomentamos la exploración global durante el comienzo de la simulación (valores superiores de *inertia weight* que producen velocidades mayores en las partículas) e intensificamos la exploración local en la fases finales de la simulación (valores inferiores de *inertia weight* que implican velocidades inferiores).

La asignación de valores de los factores de aceleración o ratios de aprendizaje  $\varphi_1$  y  $\varphi_2$  la realizamos según [34, 56], donde se recomienda  $\varphi_1 = \varphi_2 = 2$ , para igualar los efectos de los factores cognitivo y social.

El algoritmo se ejecuta en un entorno de vecindario local de tamaño 3 vecinos y con topología centralizada [8]. El vecindario local se caracteriza con respecto a un vecindario global por una diversificación mayor seguida de una convergencia más lenta.

Los primeros resultados del algoritmo PSO aplicado al problema MNDP no fueron tan buenos como esperábamos. Para mejorarlos incluimos un método de búsqueda local similar al del algoritmo evolutivo. Esta operación no se plantea en todas las iteraciones evitando una excesiva carga computacional. Consiste fundamentalmente en la eliminación/re-ubicación de estaciones base prescindibles. Utilizamos estas técnicas una de cada tres iteraciones en el caso de la eliminación y una de cada seis en el caso de la re-colocación de elementos. Una continúa eliminación de estaciones base en cada partícula supone una reducción significativa en el número medio de estaciones a lo largo de la simulación. Para compensar el efecto se añade una operación en la que surgen de forma espontánea nuevas estaciones base en la partícula. Si bien la frecuencia de ambas operaciones es la misma (cada tres iteraciones), el instante de realización es alternado, es decir, no se realizan ambas operaciones en la misma iteración del algoritmo.

## 6.5 Especificaciones para el algoritmo Harmony Search

Hemos realizado 25 simulaciones del algoritmo HS, cada una de ellas con 9800 iteraciones. El número de iteraciones elegido resultó muy elevado, como puede observarse en la Figura 6.3, solo se aprecia una ligera mejoría de valor de *fitness* en las soluciones halladas entre las iteraciones 5000 y 9800. No obstante mantuvimos el número de iteraciones en 9800 para igualar el número de evaluaciones por simulación entre los distintos algoritmos.

Las probabilidades que determinan si se genera una melodía aleatoria o bien se realiza un ajuste de tono *pitch adjustment* varían linealmente según la iteración que se encuentre el algoritmo. En las primeras iteraciones se potencia la búsqueda global y en las últimas se acentúa la intensificación en zonas próximas a las soluciones ya encontradas. Los intervalos empleados para las probabilidades *harmony memory considering rate* HMCR y *pitch adjusting rate* PAR son los recomendados en [108], que a continuación indicamos:

$$PAR \in [0, 1 - 0, 5] \text{ con } PAR = 0, 1 \text{ en la primera iteración}$$

$$HMCR \in [0, 75 - 0, 99] \text{ con } HMCR = 0, 75 \text{ en la primera iteración}$$

En los casos en los que se realiza el ajuste *pitch adjustment*, se realiza según la Ecuación 2.12, con  $\epsilon$  obtenido a partir de la distribución uniforme  $u(-1, 1)$ . Hemos utilizado un ancho de banda para el campo de localización de la estación base de  $bw=10$ . Para el campo que identifica el tipo de la estación base, el ancho de banda empleado es  $bw=4$ . Este ajuste lo aplicamos el 33,3% de las ocasiones solo a la posición de la estación base, el 33% de las veces exclusivamente al tipo de estación y en el tercio restante de las ocasiones alteramos ambos aspectos de la estación base correspondiente. También se considera la posibilidad de que la estación base afectada sea completamente retirada de la solución o bien que aparezca una nueva estación base en la solución, ambos casos con probabilidad aproximada de 13,3%.

## 6.6 Especificaciones para el algoritmo CRO

Al igual que en el algoritmo evolutivo, con el CRO hemos realizado 60 simulaciones, cada una formada por 200 iteraciones. Una diferencia básica con respecto al evolutivo es que éste último mantiene constante el tamaño de la población en 50 individuos, mientras que el algoritmo CRO se caracteriza por una población variable habitando una estructura de coral con una determinada capacidad máxima. La capacidad máxima la fijamos como  $N_{ind} = N \times M = 7 \times 8 = 56$  individuos. Los resultados obtenidos muestran que con este tamaño de coral la población media es de 53,01 individuos. Una batería de 60 simulaciones anteriores con un arrecife de tamaño

$N_{ind} = N \times M = 10 \times 10 = 100$  ofrecieron resultados extremadamente similares en cuanto al valor de *fitness* de los mejores individuos. Por esta razón nos pareció que la diferencia de 53,01 individuos en media por iteración con respecto a los 50 individuos permanentes de los otros algoritmos no influiría en los resultados. El factor de ocupación del arrecife en el inicio del algoritmo es  $\rho = 70\%$ .

Como se desarrolló en el Capítulo 5, algunas funciones son comunes con el algoritmo evolutivo. Este es el caso de la generación de nuevos individuos aleatorios, en el sentido de que el número de estaciones base para ellos es seleccionado al azar un dentro del mismo intervalo para ambos algoritmos ( $[N_{MinB} - N_{MaxB}] = [9 - 25]$ ). Otra función común a ambos es la búsqueda local, compartiendo los criterios que califican a una estación base como prescindible: el parámetro  $\beta$  toma el mismo valor  $\beta = 0,5\%$ . En el CRO, se eliminan el 75% de las estaciones prescindibles y desplazan el 25% restante a nuevas ubicaciones.

A continuación exponemos los valores numéricos asignados a cada uno de los parámetros específicos del CRO, analizando cada función del mismo. Respecto a la reproducción sexual, la suma conjunta de las probabilidades de reproducción sexual externa e interna superan la unidad, de modo que algunos individuos aleatorios se les permite realizar simultáneamente los dos tipos de reproducción sexual. Por un lado, la probabilidad de reproducción sexual externa es del  $Pb_{ext} = 0,9$ , es decir, el 90% de los corales realizan este tipo de reproducción repartidos equitativamente entre los dos métodos de cruce explicados en el capítulo 4: TBC (Template-Based Crossover) y DMC (Direct Merge Crossover). Recordamos aquí que la reproducción sexual interna consiste en generar un descendiente a partir de dos progenitores. De este modo, la operación sobre el 90% de la población provocará un incremento de la misma en un 45%.

En cuanto a la reproducción sexual interna, le asignamos una probabilidad de  $Pb_{int} = 0,6$ . Como ya explicamos, esta operación es similar a la mutación del algoritmo evolutivo. Consiste en generar una copia del progenitor sobre la que se modifican estaciones base al azar. A efectos prácticos, esta operación puede describirse como un cruce del progenitor con otro nuevo y de creación aleatoria. En el CRO, el 70% de los cruces con un progenitor aleatorio a tal efecto son de tipo TBC. Optamos por realizar el 30% restante con cruces de tipo DMC dado que este tipo de cruce intensifica de forma general el número de estaciones base del descendiente, ofreciendo mayor diversidad a las soluciones tratadas.

La probabilidad de reproducción asexual  $Pb_{asex}$  es variable según la diversidad de la población de cada iteración. Recordemos que esta operación consiste en la clonación de uno de los mejores individuos (el 5% de los individuos con mejor valor de *fitness*) sobre el cual se altera una única estación base. Cuando en la población ya existen muchas copias (población con reducida diversidad) no interesa fomentar este tipo de operación. El umbral de diversidad en la población para permitir este tipo de reproducción lo hemos fijado en el 20%. Este valor significa que el 20% de los corales son diferentes entre sí y que el resto son copias de los primeros. Si la diversidad en la población supera el 20% la probabilidad de reproducción asexual es del  $Pb_{asex} = 5\%$  salvo en las fases de eliminación de réplicas que se incrementa hasta un 15%.

El proceso de modificación de una estación base del individuo que se reproduce de forma asexual es el siguiente: 33% de las modificaciones consisten en la eliminación de la estación base que cubre menos ubicaciones, el 33% siguiente alteran aleatoriamente el tipo de dicha estación base y el tercio restante de las ocasiones se asigna un tipo de estación aleatorio a una estación base elegida al azar.

Como ya hemos mencionado la probabilidad de reproducción asexual se incrementa intencionadamente ( $Pb_{asex} = 15\%$ ) en las iteraciones en las que se eliminan réplicas. La eliminación de réplicas se realiza cuando el algoritmo no progresa durante al menos 5 iteraciones consecutivas. Entendemos que el algoritmo no progresa de una iteración a la siguiente si no encuentra

ningún individuo con mejor valor de *fitness* al que ya disponía en la iteración anterior. Entonces se eliminan todas las réplicas de la población permitiendo un máximo de dos individuos exactos. En estos casos se incrementa la probabilidad de reproducción asexual con el objetivo de compensar la eliminación de réplicas que no hacen progresar al algoritmo con la aparición de algunas “réplicas modificadas” solo de los mejores individuos (entre el 10% de los individuos más óptimos).

El efecto de la depredación está definido por la probabilidad de depredación  $Pb_{pred} = 15\%$  y el porcentaje de individuos considerados como los peores, siendo éste  $p=5\%$ . En las fases de eliminación de réplicas estos valores aumentan considerablemente con la intención de sacar al algoritmo de un posible estancamiento, situándose en  $Pb_{pred} = 50\%$  y  $p=10\%$ .

Por último, en la operación de asentamiento de nuevos corales en el arrecife hemos asignado un número máximo de tres intentos por individuo ( $N_{Ch} = 3$ ). Si tras dichas oportunidades el individuo no ha logrado un espacio en el coral, es descartado para la siguiente iteración.

## 6.7 Resultados con los algoritmos desarrollados

### 6.7.1 Resultados obtenidos con el EA

Para poder realizar la comparación de los diferentes algoritmos se estudian las Tablas 6.8 y 6.9 junto con las Figuras 6.6 y 6.7. La Tabla 6.8 se incluyen los dos mejores individuos para cada algoritmo excepto para el *Greedy* (es el que alcanza soluciones de peor calidad). Para cada individuo se señalan los valores de los parámetros de la función de evaluación, esto es incremento de campo eléctrico, coste final y porcentaje de cobertura sobre la zona de interés, y el número total de estaciones base. El número de estaciones base es importante considerarlo porque ofrece información adicional a la que entrega la función de evaluación. La Tabla 6.9 proporciona información de los valores medios de los mismos parámetros para el conjunto de simulaciones. En ambas tablas aparecen los individuos en orden decreciente de su valor de *fitness*, reservando las últimas posiciones de las tablas para los algoritmos que mejor respuesta proporcionan. El orden de aparición de algoritmos en las tablas lo mantenemos en las figuras para facilitar la comparación de resultados con los métodos implementados.

Las Figuras 6.6 y 6.7 permiten comparar también los cinco algoritmos entre sí. La primera incluye la representación gráfica de los individuos de la Tabla 6.8 y la Figura 6.7 agrupa las evoluciones de la función de evaluación para las simulaciones ganadoras con los diferentes algoritmos, es decir, las simulaciones que determinan los mejores individuos para cada algoritmo, a excepción del algoritmo *Greedy* que no es evolutivo. Por último, para cada algoritmo incluimos figuras específicas que amplían información del mejor individuo localizado.

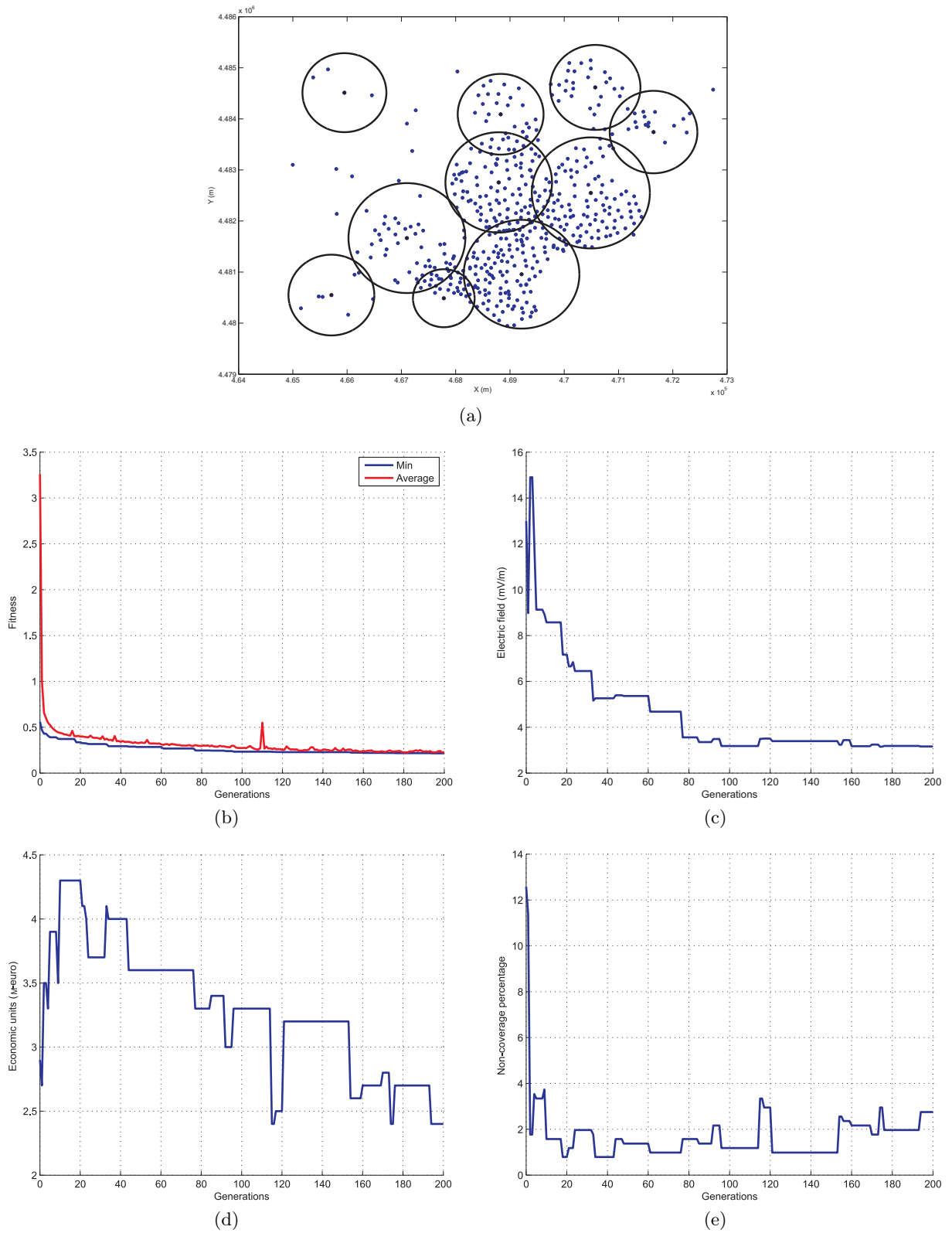
Tras realizar 60 simulaciones aleatorias y consecutivas con el algoritmo evolutivo basadas en los datos expuestos anteriormente, los resultados obtenidos se muestran en las Tablas 6.8 y 6.9, y en las Figuras 6.1, 6.6(e), 6.6(f) y 6.7(c). El algoritmo evolutivo se encuentra en la segunda mejor posición de todos, antecedido por el algoritmo CRO.

Los individuos ofrecidos por el algoritmo evolutivo con mayor grado de adaptación, y por lo tanto con menor valor de la función de evaluación, tienen un valor de *fitness* de 0,21696 y 0,21554, con 11 y 10 estaciones base respectivamente. El porcentaje de cobertura en ambos es superior al 97% y el incremento de campo eléctrico es 3,53 mV/m en la primera solución y 3,16 mV/m en la mejor de ambas. Las representaciones gráficas para estos individuos aparecen en las Figuras 6.6(e) y 6.6(f), respectivamente. En ambas se incluyen los puntos de interés (conjunto  $\mathcal{P}'$ ) para el servicio de red de telecomunicación y alcance de cobertura de las estaciones base de dicha red. El porcentaje de no cobertura se obtiene a partir del cociente entre el número de

puntos de  $\mathcal{P}'$  que quedan fuera de cualquier círculo en la figura y el cardinal de  $\mathcal{P}'$ . La diferencia de tamaño de las circunferencias se debe a que el radio de cobertura de una antena depende de su potencia de transmisión y de la altura a la que se haya instalado. Dado que las altitudes de las posibles ubicaciones para las antenas (conjunto  $\mathcal{P}$ ) son muy similares entre sí, la diferencia en la longitud de los radios se debe fundamentalmente a la potencia de transmisión. Comparando las Figuras 6.6(e) y 6.6(f) podemos comprobar que el mejor individuo deja un número de puntos sin atender mayor que la segunda mejor solución, a costa de mejorar en los otros dos parámetros objetivo de la función de evaluación (incremento de campo eléctrico y coste de la red), como refleja la Tabla 6.8. La Figura 6.6(f) se muestra ampliada en la Figura 6.1(a), a la cual se añaden el resto de subfiguras que muestran diferentes aspectos de la evolución de los parámetros objetivo a través de las 200 generaciones de la simulación. La Figura 6.1(b) (coincidente con la Figura 6.7(c)) representa las curvas de los valores medio (en color rojo) y mínimo (en color azul) de la función de *fitness* de la población en cada generación. Los valores de estas curvas en la última generación (generación número 200) coinciden con los datos incluidos en las Tablas 6.8 y 6.9 respectivamente. Destaca en esta figura cómo la curva para el valor medio de la función de evaluación alcanza ciertos picos que son debidos a la aparición de nuevos individuos de peor calidad que la media, pero que sin embargo aportan diversidad a la población. Por otro lado, comprobamos que la población no ha convergido en la última generación todavía, al no cumplir la Ecuación 6.3 con  $\epsilon = 1\%$ .

La curva del mínimo valor de *fitness* de la Figura 6.1(b) se desglosa en tres curvas, una para cada parámetro protagonista de la función de evaluación: incremento de campo eléctrico en la Figura 6.1(c), coste de la red en la Figura 6.1(d), y porcentaje o grado de no cobertura sobre el área de estudio  $A$  en la Figura 6.1(e). Como puede apreciarse la tendencia de las tres gráficas es descendiente, aunque en ciertas generaciones se incrementa el valor de alguno de ellos si esto supone reducir el valor final de la función de *fitness* (con una mayor reducción en alguno de los otros dos parámetros que permite mejorar el valor de *fitness*).

Finalmente, los valores promedio de las 60 simulaciones realizadas con el algoritmo evolutivo se resumen en la Tabla 6.9, donde se tiene un valor de *fitness* medio de 0,2407, un incremento medio de campo eléctrico de 4,3 mV/m, un coste medio de la red de  $2,862 \cdot 10^6 \text{€}$  y un porcentaje de no cobertura menor del 2%. El algoritmo evolutivo junto con el CRO, son los que ofrecen soluciones con menor número de estaciones base.



**Figura 6.1:** Características de la mejor solución ofrecida por el algoritmo EA: (a) representación gráfica del individuo, (b) evolución de la función de evaluación de la simulación que genera la solución, (c) incremento de campo eléctrico para este individuo, (d) coste de la red y (e) porcentaje del área de estudio  $A$  que queda sin servicio.

**Tabla 6.8:** Mejores individuos obtenidos con cada uno de los algoritmos implementados. Los individuos aparecen ordenados de mayor a menor valor de *fitness*: algoritmo *Greedy*, PSO, HS, EA y CRO. (\*\*) Corresponden a las segundas mejores soluciones obtenidas por sus correspondientes algoritmos. La representación gráfica para todos ellos puede verse en la Figura 6.6.

Algoritmo	<i>fitness</i>	$\Delta E(\text{mV/m})$	$\mathcal{C}(\cdot 10^6 \text{€})$	% No cobertura	Núm. estaciones
Greedy	0,36871	5,72	2,7	8,45	12
PSO	0,28933	5,00	3,6	1,96	18
HS(**)	0,22703	3,49	2,6	2,55	17
HS	0,22355	3,74	2,8	1,57	20
EA(**)	0,21696	3,53	2,8	1,38	11
EA	0,21554	3,16	2,4	2,75	10
CRO(**)	0,18985	3,36	2,1	3,14	9
CRO	0,18529	3,54	2,3	2,16	10

**Tabla 6.9:** Valores promedio para los parámetros que caracterizan las soluciones encontradas con los algoritmos EA, *Greedy*, HS, PSO y CRO.

Algoritmo	<i>fitness</i>	$\Delta E(\text{mV/m})$	$\mathcal{C}(\cdot 10^6 \text{€})$	% No cobertura	Núm. estaciones
Greedy	0,5839	9,60	2,375	18,35	12,31
PSO	0,3387	7,00	3,793	2,52	17,83
HS	0,2375	3,99	2,848	2,02	18,56
EA	0,2407	4,34	2,862	1,86	11,42
CRO	0,2041	4,06	2,448	4,01	11,1

### 6.7.2 Resultados obtenidos con el algoritmo *Greedy*

Los resultados obtenidos con este conjunto de repeticiones se recogen nuevamente en las Tablas 6.8 y 6.9 y en la Figura 6.6(a). No se muestra ninguna gráfica de evolución para la función de *fitness* porque este algoritmo no está basado en poblaciones ni tampoco evoluciona un número determinado de generaciones, sino que en cada repetición trabaja con un único individuo que intenta optimizar. La Tabla 6.8 muestra las características del mejor individuo localizado por este algoritmo. Se trata del peor individuo de la tabla, con valores máximos en la tabla tanto para el valor de *fitness* (0,3687), como para el incremento de campo eléctrico (5,72 mV/m) como para el porcentaje de falta de cobertura (8,45 %). En la Figura 6.6(a) también se aprecian gráficamente ambas circunstancias. El elevado incremento de campo eléctrico se explica por el alto número de puntos situados en intersecciones de circunferencias, tratándose de puntos situados en la zona de cobertura de más de una estación base y por lo tanto sometidos a un campo eléctrico mayor. La figura igualmente muestra que este individuo es el que menor grado de cobertura ofrece en la zona puesto que es la solución que mayor número de puntos deja fuera del alcance de ninguna estación base.

La Tabla 6.9 confirma que este algoritmo puede ser considerado como cota para el resto, puesto que trabaja con soluciones que en media tienen baja calidad. Los valores promedio de las magnitudes referidas en la tabla son en general elevados con respecto a los valores generados por el resto de algoritmos, especialmente: el valor de *fitness* de 0,58 frente al resto de valores siempre por debajo de 0,34; el incremento de campo eléctrico de 9,6 mV/m que duplica los valores de incremento de campo de los tres últimos algoritmos en la tabla (HS, evolutivo y CRO) y el porcentaje de no cobertura de 18,35 % comparado con el resto de algoritmos resulta excesivo.



Los valores promedio para el coste total de la red y el número de estaciones son similares a los valores del resto de algoritmos en la lista.

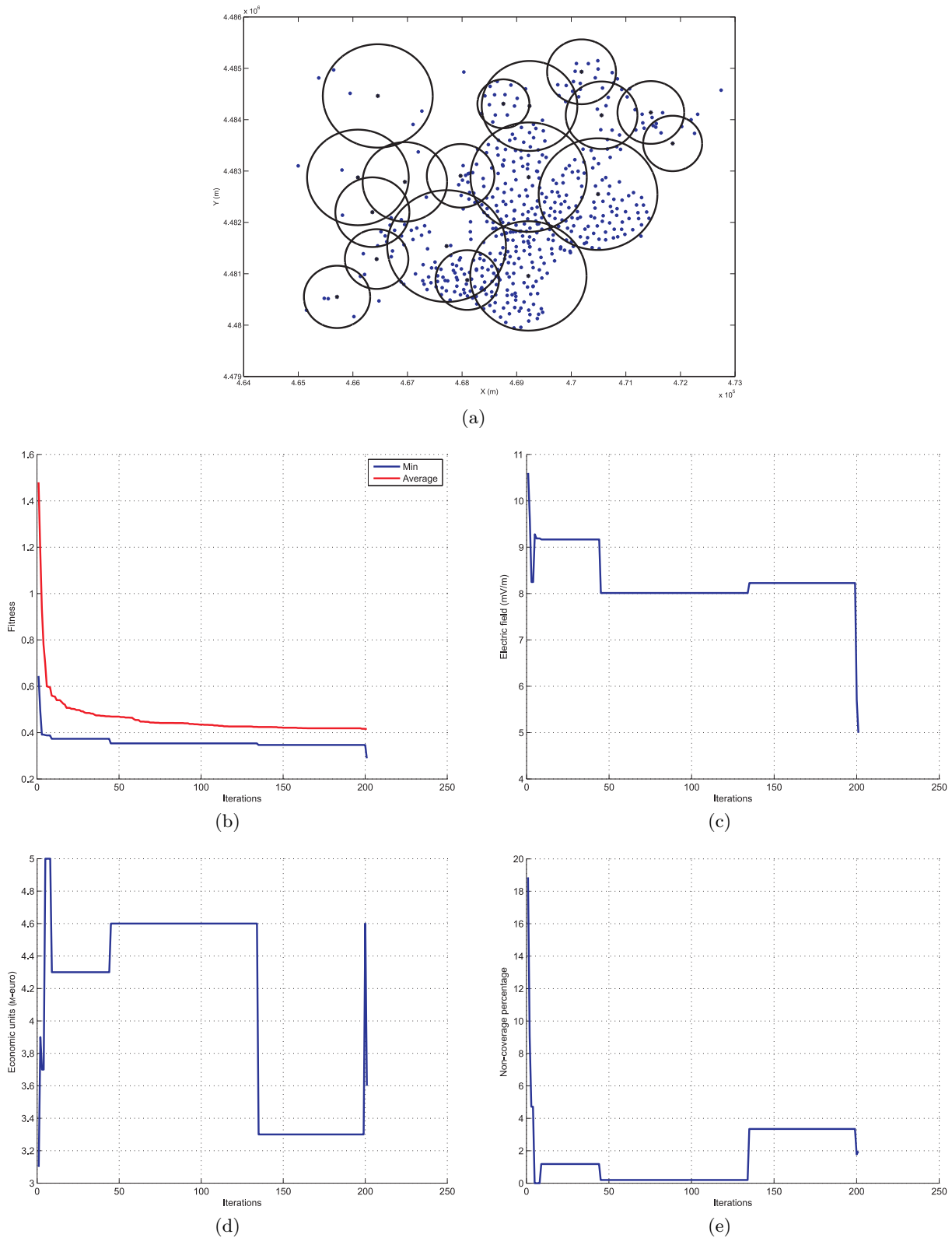
### 6.7.3 Resultados obtenidos con el algoritmo PSO

Los resultados de las simulaciones realizadas con el algoritmo PSO bajos los criterios descritos se recogen en las Tablas 6.8 y 6.9 y en las Figuras 6.2, 6.6(b), 6.7(a). Como en los casos anteriores, la Figura 6.2 recoge la representación gráfica del mejor individuo identificado (que a su vez se incluye en la Figura 6.6(b) para comparar el PSO con el resto de algoritmos) y la evolución de la función de evaluación en el progreso de la simulación (que también se añade en la Figura 6.7(a) con el mismo fin comparativo).

El segundo peor resultado de la Tabla 6.8 corresponde al algoritmo PSO, precedido solo por el algoritmo *Greedy*. El individuo obtenido con el algoritmo PSO está caracterizado por un valor de *fitness* elevado (0,28933), siendo un 56,15% superior al mejor *fitness* del algoritmo CRO en la tabla. También se caracteriza por valores elevados tanto para el incremento de campo eléctrico (5 mV/m), como para el coste estimado de la red ( $3,6 \cdot 10^6 \text{€}$ ) como para el número de estaciones base (18 estaciones). La Figura 6.2(a), coincidente con la Figura 6.6(b), representa gráficamente esta solución. En la representación destaca el elevado número de estaciones base, repercutiendo en el coste de la red, frente a otros algoritmos. Además se aprecia la existencia de numerosas zonas de intersección de las circunferencias que representan la cobertura de las estaciones base. Esto deriva en un mayor incremento de campo eléctrico total. Podemos observar al menos dos estaciones base de menor potencia pudieran ser etiquetadas como *prescindibles* al cubrir puntos que en su mayoría ya se encuentran atendidos por estaciones de mayor potencia. El porcentaje de cobertura es reducido (1,96%) dado que la zona es cubierta por la red casi en su totalidad. La Tabla 6.9 es consistente con estos comentarios, mostrando valores promedio elevados para el *fitness*, el incremento de campo eléctrico, el coste de la solución y el número de estaciones base. Se concluye por tanto que el algoritmo PSO es el menos indicado para resolver el problema MNDP después del algoritmo *Greedy*.

La Figura 6.2(b) (coincidente con 6.7(a)) muestra la evolución de la función de evaluación para la simulación que genera el mejor individuo con el PSO. Como en casos anteriores, se representan las curvas para los valores medio (color rojo) y mínimo (color azul) de la función de *fitness*. Se observa que ambas curvas tienen una tendencia decreciente alcanzando el mínimo en la última iteración. En este punto el valor de ambas curvas coinciden con la información registrada en las Tablas 6.9 y 6.8. Destaca también la ausencia de picos en el valor medio de *fitness* sugiriendo que los nuevos individuos que se incorporan a la población son de calidad media con respecto al resto en esa iteración. Podemos pensar que sería favorable potenciar la diversidad en las distintas poblaciones. Respecto a la curva del valor mínimo para el *fitness*, se observan tres zonas de pendiente nula, lo cual indica cierto estancamiento en el progreso del algoritmo sin mejorar en las soluciones halladas durante iteraciones consecutivas.

Finalmente la curva del valor mínimo de la función de evaluación de la Figura 6.2(b) se desglosa en tres gráficas, una por cada componente de la función de evaluación: el incremento de campo eléctrico en la Figura 6.2(c), el coste de la red debido a la instalación de estaciones base en la Figura 6.2(d) y la Figura 6.2(e) para el porcentaje de no cobertura sobre la zona. Estas tres gráficas corroboran la idea del estancamiento del algoritmo desde la iteración 30 a la 140 aproximadamente, debido a los tramos horizontales comunes para las tres curvas.



**Figura 6.2:** En el algoritmo PSO, características del mejor individuo obtenido mediante 60 simulaciones de poblaciones de tamaño 50 individuos: (a) representación gráfica de la solución, (b) evolución de la función de evaluación durante 200 iteraciones de la simulación que entrega el mejor individuo, (c) incremento de campo eléctrico para esta solución, (d) coste de la red y (e) porcentaje del área de estudio  $A$  sin cobertura en este individuo.

#### 6.7.4 Resultados obtenidos con el algoritmo HS

Los resultados obtenidos mediante el algoritmo HS son relativamente aceptables pero no con la calidad deseada. Como para los algoritmos anteriores los resultados se presentan en las Tablas 6.8 y 6.9 y en las Figuras 6.3, 6.6(c), 6.6(d), 6.7(b). Si bien en las tablas sugieren resultados aceptables, al observar la representación gráfica de los dos mejores individuos obtenidos con este algoritmo en las Figuras 6.6(c) y 6.6(d), se observa que resultan poco óptimos. En general utilizan un excesivo número de estaciones base para cubrir la zona de interés. El valor de *fitness* no es elevado porque muchas de estas estaciones son de baja potencia de transmisión que además de tener menor coste que otras, generan menor incremento de campo eléctrico por cubrir menor superficie de  $A$ . En la Figura 6.6 puede verse que otros algoritmos proponen soluciones mejores empleando menor número de estaciones base pero con mayor potencia de transmisión.

Las características principales de las dos mejores soluciones con el algoritmo HS se registran en la Tabla 6.8. El valor de *fitness* para estos individuos supera en un 22,5% y en un 20,6% respectivamente al valor del mejor individuo de la tabla (perteneciente al algoritmo CRO). Este aumento se debe fundamentalmente al alto número de estaciones base empleadas ya mencionado (17 y 20 estaciones en respectivas soluciones). El resto de las variables en la tabla (incremento de campo eléctrico, coste y porcentaje de localizaciones sin cobertura) si bien toman valores mayores con respecto a la mejor solución de la tabla, no destacan considerablemente<sup>1</sup>. La representación gráfica de estas dos soluciones se muestran en las Figuras 6.6(c) y 6.6(d).

En la Tabla 6.9 el algoritmo HS se sitúa en el centro de la lista ordenada según su valor de *fitness*, con valores intermedios para la mayoría de los parámetros de la tabla, a excepción del incremento de campo eléctrico y del número de estaciones base. En cuanto al incremento de campo eléctrico, el algoritmo HS ofrece los mejores resultados promedio (3,99 mV/m). Sin embargo, dado que la función de evaluación pretende optimizar tres variables, el HS no es el mejor, puesto que los algoritmos evolutivos y CRO mejoran el resto de parámetros con un incremento de campo medio similar.

La Figura 6.3 muestra con mayor detalle la mejor solución hallada por el algoritmo HS, tanto su representación gráfica en la Figura 6.3(a) (coincidente con la Figura 6.6(d)), como la evolución de la función de evaluación de la simulación que genera dicha solución en la Figura 6.3(b) (también en la Figura 6.7(b)). La curva en color rojo representa el valor medio del *fitness* de las soluciones residentes en la memoria para cada iteración, mientras que la curva en color azul indica el valor de la función de evaluación para la mejor melodía encontrada en la simulación, es decir, el valor mínimo del *fitness* almacenado en la memoria. Ambas curvas tienen tendencia decreciente, la primera de ellas más suave y la curva azul aparece escalonada indicando que se encuentra una mejor solución cada cierto número de iteraciones consecutivas. La curva del mínimo valor de *fitness* de esta gráfica se desglosa en las tres representaciones individuales: la Figura 6.3(c) con la evolución del incremento de campo eléctrico para la mejor solución encontrada en la simulación que nos ocupa, la Figura 6.3(d) respecto al coste de las mejores soluciones por iteración y la Figura 6.3(e) que muestra la evolución del porcentaje de no cobertura para el mejor individuo en cada iteración. En las tres aparecen fluctuaciones que van disminuyendo conforme progresa la simulación. En el intervalo de 5000 a 9800 iteraciones los cambios no son muy destacados, aunque sí apreciables.

---

<sup>1</sup>Modificando los costes de las estaciones base de la Tabla 6.2 o bien el peso de los coeficientes de la función de evaluación en la Tabla 6.5 podríamos conseguir que la función de evaluación penalizara el uso de estaciones base de baja potencia frente a la utilización de antenas de mayor potencia de transmisión.

### 6.7.5 Resultados obtenidos con el algoritmo CRO

Los resultados obtenidos bajo las condiciones arriba expuestas quedan reflejados en las Tablas 6.8 y 6.9 y en las Figuras 6.4, 6.5, 6.6(g), 6.6(h), 6.7(d). El algoritmo CRO figura en la Tabla 6.8 como el que mejores resultados obtiene, obteniendo soluciones con los mejores valores de *fitness* (0,18985 y 0,18529) de la lista, con los menores costes totales ( $2,1 \cdot 10^6 \text{€}$  y  $2,3 \cdot 10^6 \text{€}$ ) y el menor número de estaciones base por empleadas (9 y 10 estaciones). Así mismo tiene muy bajos niveles para el incremento de campo eléctrico (3,36 mV/m y 3,54 mV/m) y porcentajes de no cobertura aceptables (3,14% y 2,16%). En la Tabla 6.9 se repite este patrón, destacando que el algoritmo CRO trabaja con individuos con porcentajes de no cobertura mayores que los otros algoritmos implementados. Si deseáramos reducir dicho porcentaje sería conveniente modificar los valores de los coeficientes de la función de evaluación de la Ecuación 6.1, en la página 95, asignando un peso mayor a  $k_3$ . Para que pudieran compararse los valores de *fitness* de todos los algoritmos, habría que aplicar el mismo cambio en la función de evaluación de todos ellos.

La Figura 6.4(a) muestra la representación del mejor individuo hallado por el algoritmo CRO. Destaca la situación estratégica de las estaciones base cubriendo prácticamente todas las ubicaciones de interés y además con reducidas zonas de redundancia (más de una estación base cubriendo una localización). Incluso destaca la acertada elección del tipo de cada estación base de la solución. Esta solución también aparece en la Figura 6.6(h) junto con la segunda mejor solución hallada por el algoritmo, Figura 6.6(g), con el fin de comparar las representaciones gráficas de las mejores soluciones de todos los algoritmos reflejadas en la Tabla 6.8. Las dos soluciones del CRO son muy parecidas, diferenciándose en una única estación base situada en el cuadrante superior izquierdo del área de estudio, producto de una reproducción asexual probablemente.

La Figura 6.4(b) ofrece las curvas de evolución para el valor medio (color rojo) y el valor mínimo (color azul) de *fitness* en la simulación que produce la mejor solución. Esta figura es un extracto de la Figura 6.7(d) que recorre las 200 iteraciones. La primera representa solo las últimas 120 iteraciones con el fin de apreciar mejor las diferencias entre las curvas de valor medio y mínimo. Al igual que en los algoritmos anteriores, esta evolución de la función multiobjetivo se desglosa en tres curvas: incremento de campo eléctrico (en la Figura 6.4(c)), coste de la red (en la Figura 6.4(d)) y porcentaje de no cobertura (en la Figura 6.4(e)). Las tres curvas tienen tendencia descendente a lo largo de la simulación.

Finalmente en la Figura 6.5 mostramos la ocupación del arrecife durante cinco periodos equiespaciados de cuatro iteraciones consecutivas cada uno de ellos en la simulación que ofrece el mejor de los resultados. Representamos el arrecife como una cuadrícula bidimensional de  $N \times M = 7 \times 8$  celdas, donde el color de la celda indica si el coral ocupante es de nueva ocupación (color azul) o ya era residente en la anterior iteración (color rojo). El color negro localiza la solución más óptima del arrecife encontrada hasta ese momento, mientras que el color blanco simboliza que la celda se encuentra disponible para un nuevo asentamiento. Cada fila en la figura corresponde a un grupo de iteraciones consecutivas: la primera para las cuatro iteraciones en el inicio de la simulación, la segunda fila para las iteraciones [49-52], las dos siguientes líneas corresponden a intervalos de cuatro iteraciones separadas 50 iteraciones entre sí, es decir [99-102] y [149-152], reservando la última fila para las iteraciones finales de la simulación [197-200]. En la primera iteración, representada en la Figura 6.5(a), no aparecen casillas con color azul porque se corresponde al inicio del algoritmo con tantas celdas desocupadas como determine el factor de ocupación empleado ( $\rho = 0,7$  implica que 39 celdas queden habitadas frente a las 56 totales que componen el arrecife). La competencia por el espacio está presente desde el comienzo de la simulación puesto que ya en la segunda iteración aparecen numerosas celdas con color azul. La cantidad de casillas azules se reduce conforme evoluciona la simulación, indicando que los

individuos son cada vez mejores y resisten durante más tiempo en el espacio conquistado. Puede observarse cómo se encuentra una nueva solución mejor en cada iteración de la primera fila, variando por tanto la posición de la celda de color negro, dado que se comienza con soluciones aleatorias y el algoritmo progresa rápidamente. En cada una de las siguientes filas no ocurre esta circunstancia, manteniéndose el coral ganador durante iteraciones sucesivas.

El salto de la iteración número 49 a la 50 (Figuras 6.5(e) y 6.5(f)) puede representar un ejemplo de depredación, liberando huecos en el arrecife para soluciones futuras. Otro detalle que podemos ver en esta figura es la aparición de lo que podría ser una réplica del mejor individuo. Esto tiene lugar en la iteración número 52, en la Figura 6.5(h). Por último queremos mencionar la mejor solución permanece ganadora al menos desde la iteración 149 en la Figura 6.5(m) hasta el fin de la simulación en la Figura 6.5(t), no variando la situación de la celda de color negro. Esto resulta coherente con lo representado en las Figuras 6.4(b) y 6.7(d), donde la curva para el mínimo valor de *fitness* tiene pendiente nula desde alguna iteración anterior a la número 140.

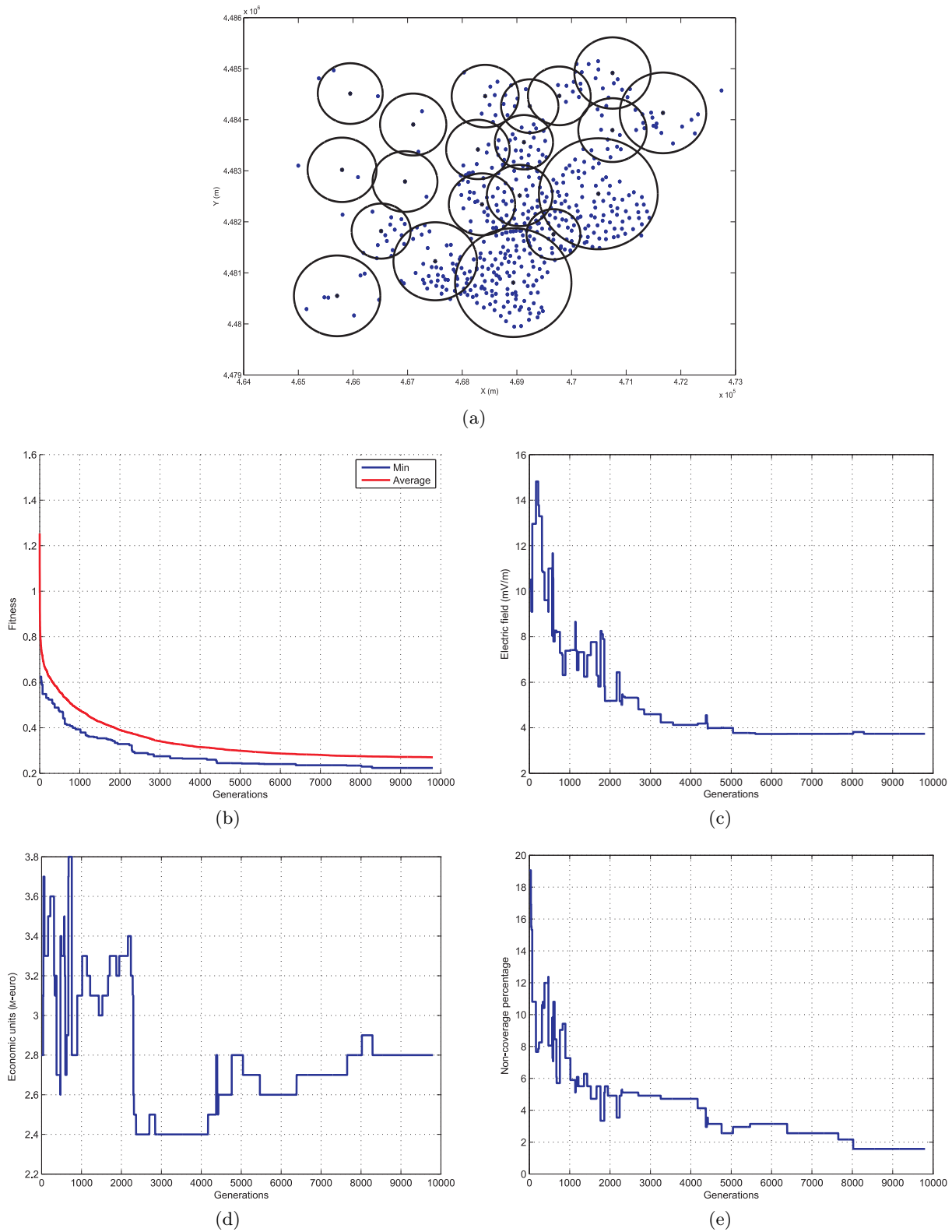
## 6.8 Comparación de las soluciones encontradas

Esta sección vamos a comparar brevemente las mejores soluciones halladas por los distintos algoritmos utilizados en este trabajo y resumidos en las Figuras 6.6 y 6.7. En la Figura 6.6 observamos la representación gráfica de las soluciones indicando localizaciones de interés y cobertura de las estaciones base. El algoritmo *Greedy* de la Figura 6.6(a) deja un número destacado de puntos sin servicio, mientras que los algoritmos PSO, en la Figura 6.6(b), y HS de las Figuras 6.6(c) y 6.6(d), emplean numerosas estaciones base de menor potencia de transmisión identificadas en las imágenes como circunferencias de menor radio. Son soluciones por tanto con un número elevado de estaciones base frente al resto de algoritmos. En esta comparación de resultados apreciamos que las soluciones más óptimas son ofrecidas por los algoritmos evolutivo y CRO. Destacamos la similitud entre los individuos representados en las Figuras 6.6(f) (EA) y 6.6(h) (CRO), con igual número de estaciones base y disposición de las mismas. El algoritmo CRO logra superar al evolutivo variando ligeramente el tipo y ubicación de cuatro estaciones base y con ello reduciendo el valor de *fitness* en un 14%<sup>1</sup>.

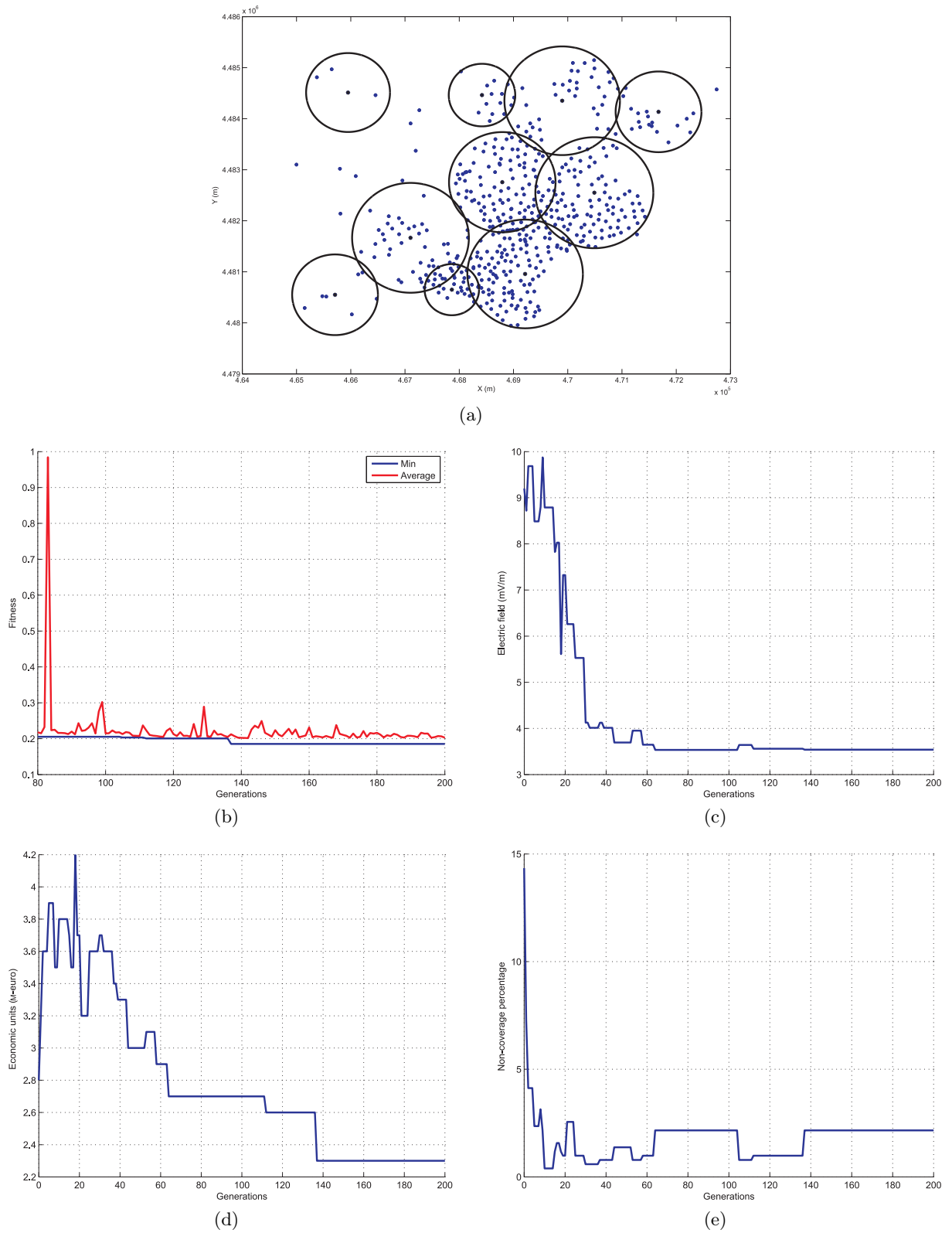
Por último en la Figura 6.7 podemos apreciar cómo en las primeras iteraciones o generaciones de todos los algoritmos existen individuos de muy baja calidad que hacen que el valor medio de *fitness* sea elevado. Para todos los algoritmos este valor medio de la función de evaluación desciende rápidamente según progresan las iteraciones. En los algoritmos PSO y HS, en las Figuras 6.7(a) y 6.7(b) respectivamente, la curva del valor medio de *fitness* aparece sin altibajos al contrario que ocurre en los algoritmos evolutivo y CRO (Figuras 6.7(c) y 6.7(d) respectivamente), lo cual resulta muy positivo tanto para favorecer la diversidad como para escapar de óptimos locales.

---

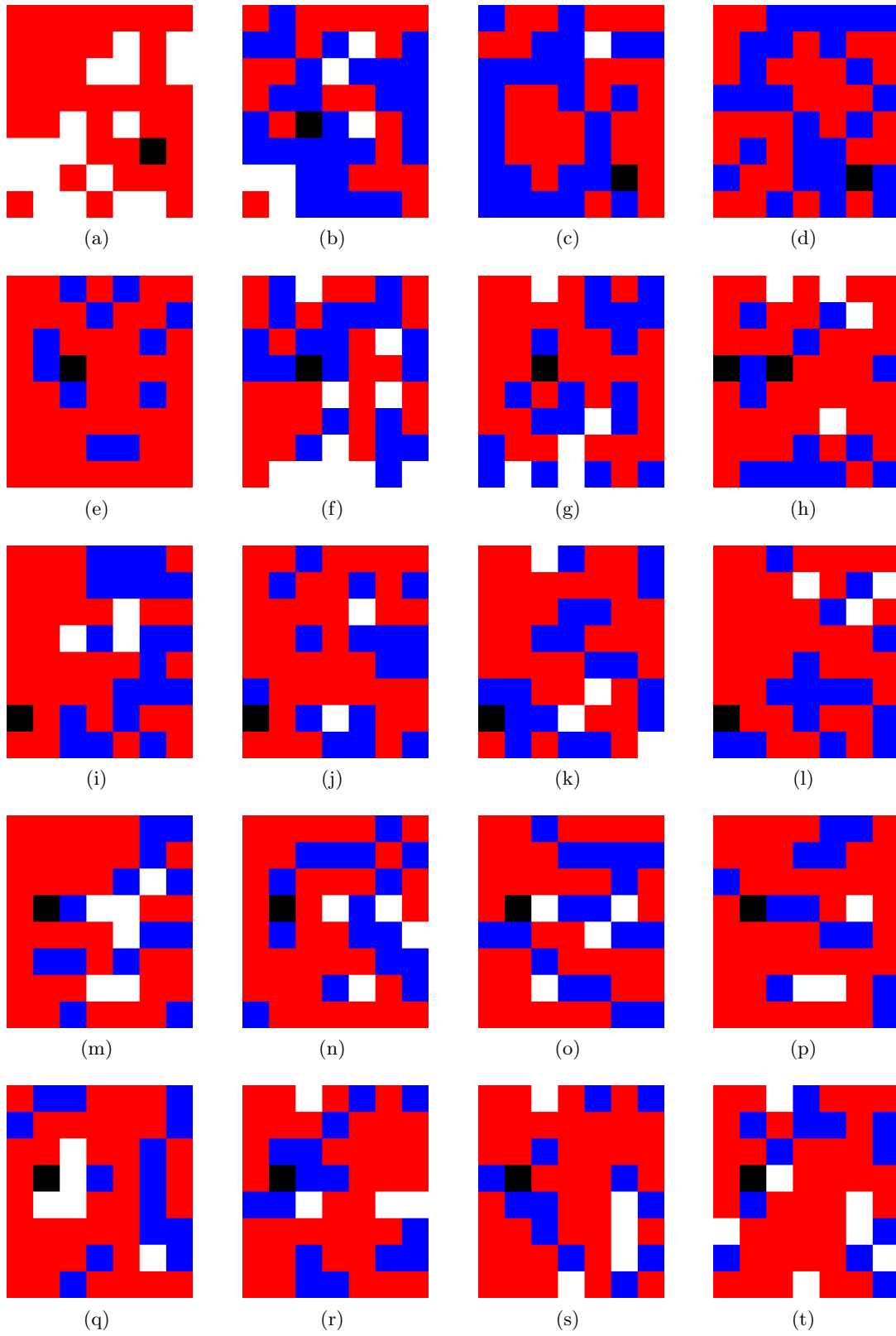
<sup>1</sup>Este porcentaje se obtiene de la comparación en la Tabla 6.8 de las soluciones ofrecidas por los algoritmos CRO y EA con los valores de *fitness* más bajos.



**Figura 6.3:** En el algoritmo HS, evolución durante 9800 iteraciones de las tres variables de la función de *fitness*: 6.3(c) incremento de campo eléctrico, 6.3(d) coste de la red y 6.3(e) porcentaje del área de estudio  $A$  sin cobertura.

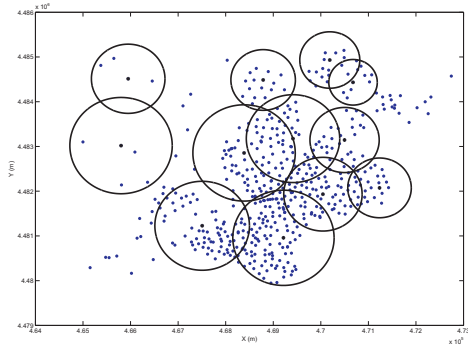


**Figura 6.4:** En el algoritmo CRO, evolución durante 200 iteraciones de las tres variables de la función de *fitness*: 6.4(c) incremento de campo eléctrico, 6.4(d) coste de la red y 6.4(e) porcentaje del área de estudio  $A$  sin cobertura.

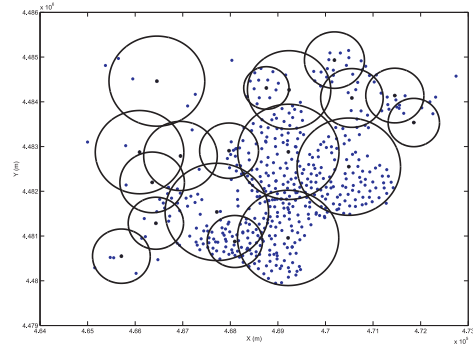


**Figura 6.5:** En el algoritmo CRO, evolución de la ocupación del arrecife coral durante 200 iteraciones en cinco periodos equiespaciados de cuatro iteraciones consecutivas cada uno de ellos.

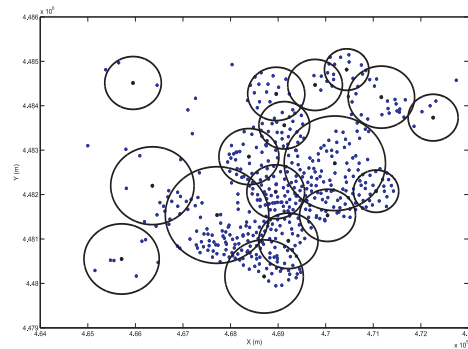




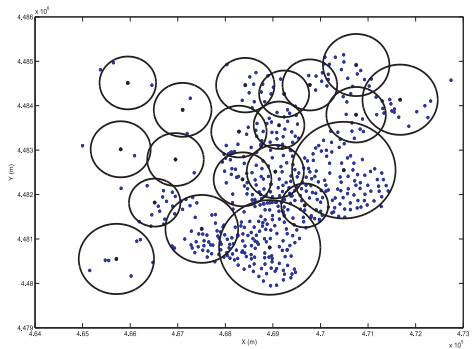
(a) Representación gráfica del individuo final resuelto por el algoritmo *Greedy* ( $fitness = 0,3687$ )



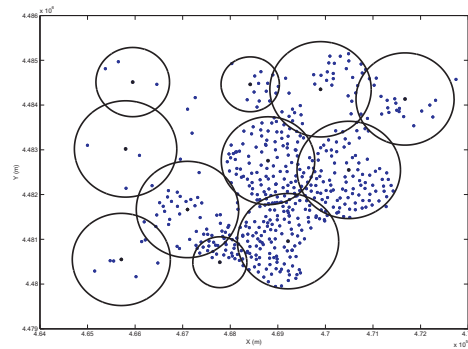
(b) Representación gráfica del individuo final resuelto por el algoritmo PSO ( $fitness = 0,28933$ )



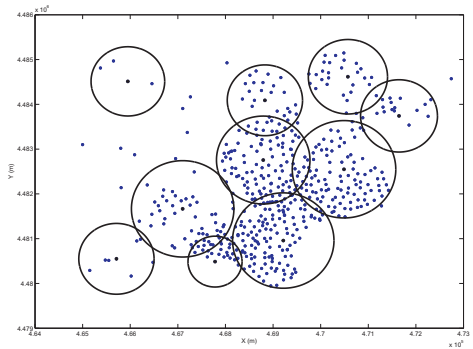
(c) Representación del segundo mejor individuo con el algoritmo HS ( $fitness = 0,2270$ )



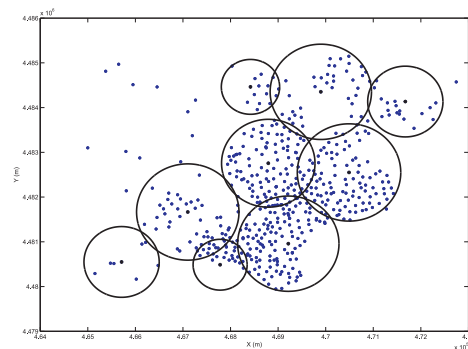
(d) Representación de la mejor solución ofrecida por el algoritmo HS ( $fitness = 0,2235$ )



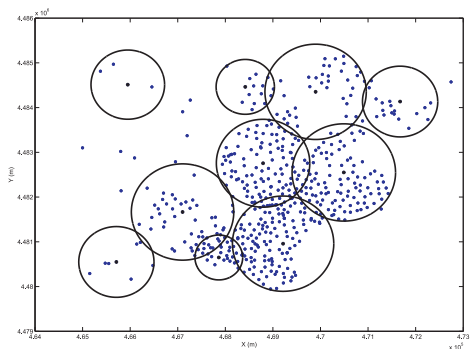
(e) Representación del segundo mejor individuo obtenido con el EA ( $fitness = 0,2170$ )



(f) Representación del mejor individuo hallado con el EA con  $fitness = 0,2155$

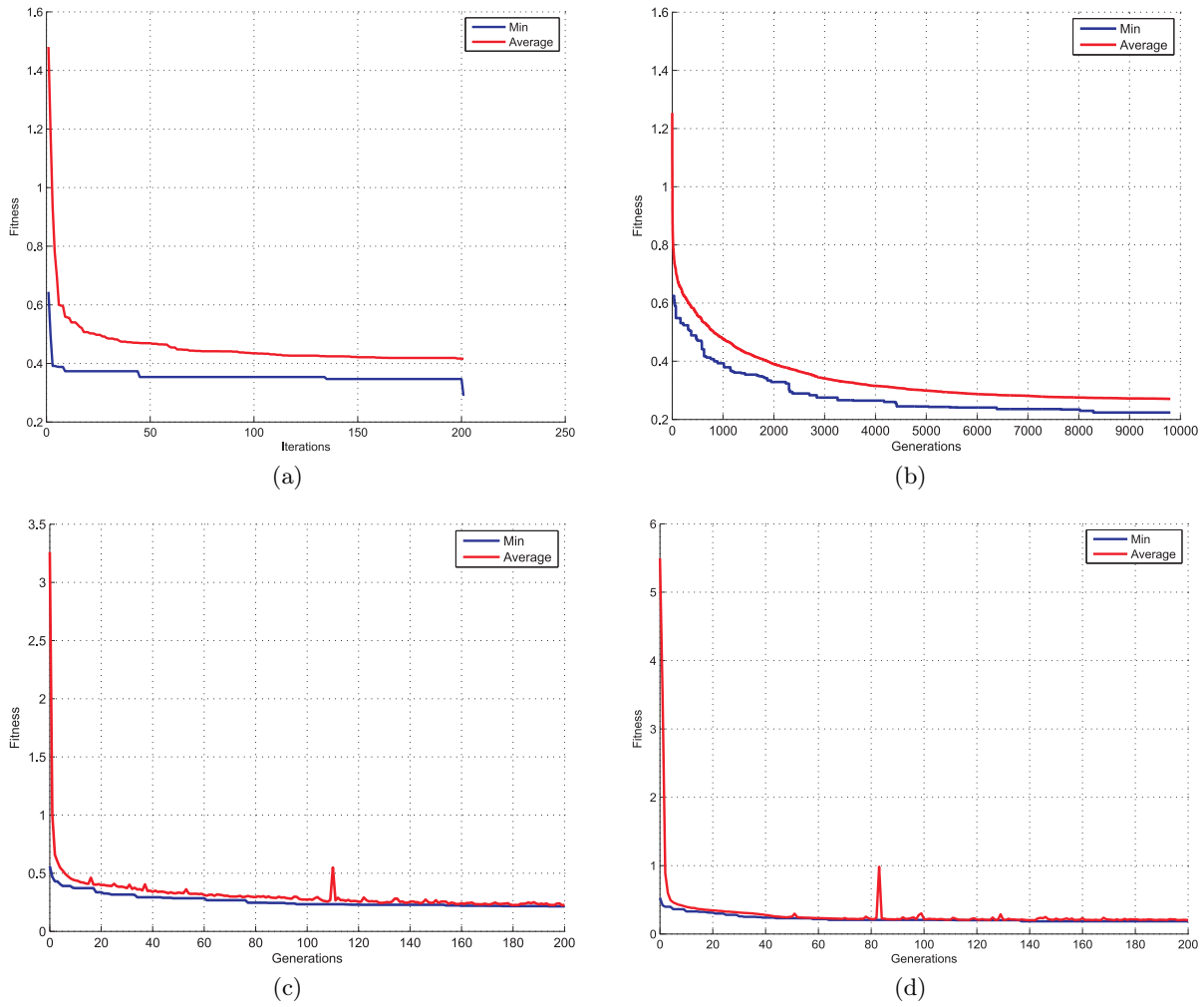


(g) Representación del segundo mejor individuo obtenido mediante el algoritmo CRO ( $fitness = 0,18985$ )



(h) Representación del mejor individuo hallado con el algoritmo CRO, valor de  $fitness = 0,18529$

**Figura 6.6:** Representación gráfica de los mejores individuos ofrecidos por los algoritmos desarrollados (ver Tabla 6.8 para mayor detalle): algoritmo *Greedy* en (a); algoritmo PSO en (b); algoritmo de búsqueda armónica en (c) y (d); algoritmo evolutivo en (e) y (f); y algoritmo CRO en (g) y (h).



**Figura 6.7:** Evolución de la función de evaluación (valor medio en color azul y valor mínimo para el mejor individuo en color rojo) en los diferentes algoritmos: (a) algoritmo PSO; (b) algoritmo HS; (c) algoritmo EA; (d) algoritmo CRO.

## Capítulo 7

# Conclusiones

En este último capítulo revisamos los objetivos planteados en la Sección 1.4, señalando las principales aportaciones del trabajo. También planteamos las líneas futuras con las que continuaremos la investigación, una primera versión de la aplicación del problema MNDP con otro algoritmo ya se ha desarrollado y la hemos incluido en el Anexo B. Cerramos el capítulo con la relación de publicaciones a las que ha dado lugar este trabajo.

### 7.1 Aportaciones de la Tesis Doctoral

Como primera aportación destacamos la incorporación de la variable radiación electromagnética en la función de evaluación de los algoritmos de optimización desarrollados para el problema MNDP. Como comentamos en la Sección 2.5, no hemos encontrado en la literatura estudios en los que se trate de minimizar la polución electromagnética sobre el terreno. Primero se desarrolló un algoritmo evolutivo (EA) con una función de *fitness* definida por tres variables y cuyo interés era minimizarla: coste económico de la red, grado o porcentaje de cobertura sobre las localizaciones objetivo e incremento de campo eléctrico en la zona de estudio. Los resultados fueron mejores que con los algoritmos con los que se comparó el EA: *Greedy*, PSO y HS.

La principal aportación de la tesis es la aplicación del algoritmo CRO al problema MNDP. El CRO es un nuevo algoritmo metaheurístico publicado recientemente [88, 89]. Los experimentos realizados demuestran la eficacia del CRO frente al resto de algoritmos (*Greedy*, PSO, HS, EA) en la resolución del problema de localización de terminales en una red de telecomunicación. Las Tablas 6.8 y 6.9 de la página 106 confirman claramente esta afirmación, de hecho consideramos que dichas tablas recogen esquemáticamente los resultados de este documento. De ellas se puede deducir que los algoritmos ordenados de mejor a peor en la aplicación en el problema MNDP son: CRO, EA, HS, PSO y *Greedy*. En la Tabla 6.8 puede comprobarse que la mejor solución hallada con el algoritmo CRO tiene un valor de *fitness* inferior al menos en un 11,92% al valor de *fitness* de las mejores soluciones generadas por el resto de algoritmos <sup>1</sup>. De entre todas las simulaciones referidas en la Tabla 6.9, el valor medio de *fitness* de las soluciones encontradas por el CRO es inferior en un 15,21% al valor medio del segundo mejor algoritmo de la lista <sup>2</sup>.

---

<sup>1</sup>Este porcentaje ha sido obtenido al comparar la solución de la Tabla 6.8 identificada como CRO(\*\*) con la solución EA.

<sup>2</sup>Este porcentaje se obtiene a partir de los valores de *fitness* de las dos últimas filas de la Tabla 6.9.

## 7.2 Líneas de trabajo futuras

Las líneas de investigación que surgen para continuar este trabajo son numerosas. Queremos destacar dos principalmente: mejora del proceso de búsqueda local en los distintos algoritmos y la implementación de un algoritmo genético de agrupamiento o *grouping genetic algorithm* (GGA) que permita añadir la restricción de capacidad en las estaciones base de la red.

El proceso de búsqueda local puede potenciarse en varios aspectos. En este trabajo se ha decidido desplazar o eliminar las estaciones de una solución que fueran “prescindibles”. Recordamos que una estación base era clasificada como estación prescindible si daba servicio a un número de localizaciones inferior a un mínimo establecido o bien todas las localizaciones que atendía estaban a su vez en la zona de cobertura o influencia de alguna otra estación base presente en la solución. Podemos por tanto ampliar la búsqueda estudiando estaciones base no prescindibles, buscando un desplazamiento de éstas que mejore el valor de *fitness* de la solución. El desplazamiento puede estar orientado a mover las estaciones base más periféricas del área de estudio  $A$  hacia el exterior, para así cubrir localizaciones limítrofes que hubiesen quedado sin servicio. Esta situación aparece en las soluciones propuestas en la Figura 6.6 de la página 115. En todas las soluciones de esta figura hay una o varias localizaciones sin servicio situadas en la parte superior derecha del área  $A$ . Colocar una nueva estación base no resulta rentable, pero sí se puede estudiar en algún caso el desplazamiento hacia la derecha de una o varias estaciones base para mejorar el porcentaje de cobertura global en  $A$ . En algunos casos podría mejorarse la solución modificando el tipo de estación con o sin desplazamiento.

Otra línea de ampliación del trabajo es considerar disjuntos los conjuntos de puntos o localizaciones  $\mathcal{O}$ ,  $\mathcal{P}$  y  $\mathcal{P}'$ . Recordamos que en la formulación matemática del problema MNDP, en la página 50, indicamos la coincidencia de estos tres conjuntos. Sin embargo, puede suceder que no dispongamos de medidas de campo eléctrico (hablamos por tanto del conjunto  $\mathcal{O}$ ) en todos los puntos integrados en los otros dos conjuntos. Además, en la vida real no sucede siempre que en una localización demandante de servicio (elemento del conjunto  $\mathcal{P}'$ ) sea factible la instalación de equipos de la red (conjunto  $\mathcal{P}$ ). Esto es debido a veces a que el lugar resulta inaccesible o poco idóneo para dicha instalación. En otras ocasiones puede ser incluso la decisión de una comunidad de vecinos con cierta aversión a los efectos para la salud que pudiera tener la radiación emitida por esos equipos. En el Anexo B se presenta una primera versión de un algoritmo en el que tiene en cuenta la no coincidencia de los conjuntos  $\mathcal{O}$ ,  $\mathcal{P}$  y  $\mathcal{P}'$ .

Una limitación real e importante es que una estación base puede atender a un número finito de demandas, determinado éste por la capacidad total disponible de dicha estación. Hasta ahora en la función de evaluación no hemos considerado la demanda no satisfecha por falta de capacidad en las estaciones, sí se consideran aquellas localizaciones que quedan fuera del área de influencia de todas las estaciones base de la solución. Sin embargo puede suceder que alguna estación base no satisfaga la demanda total de las localizaciones bajo su cobertura, y por tanto la demanda no satisfecha calculada en la Ecuación 3.1 de la página 50 resulte inferior a la real. Hemos iniciado el desarrollo de un algoritmo genético de agrupamiento que tiene en cuenta la capacidad real asignable de cada estación base. El Anexo B describe brevemente las características de dicho algoritmo. Queda para el futuro próximo la mejora y comparación con algoritmos similares.

Por último, indicar que tenemos abierta la línea de utilización de algoritmos de evolutivos multiobjetivo y su consiguiente comparación con los resultados obtenidos por los algoritmos desarrollados en el presente trabajo.

## 7.3 Lista de publicaciones

Revistas internacionales:

1. P. García-Díaz, S. Salcedo-Sanz, J. A. Portilla-Figueras and S. Jiménez-Fernández. "Mobile network deployment under electromagnetic pollution control criterion: an evolutionary algorithm approach". *Expert Systems with Applications*. Vol. 40, núm. 1. pp. 365-376, 2013. **JRC: 2,908**
2. P. García-Díaz, S. Salcedo-Sanz, J. A. Portilla-Figueras and D. Núñez-Clemente. "GSM-SIM: An educational simulation tool for teaching GSM-based mobile communications in laboratory lectures". *International Journal of Electrical Engineering Education*. Vol. 46, núm. 3 pp. 259-279, 2007. **JRC: 0,106**
3. S. Salcedo-Sanz, P. García-Díaz, J. A. Portilla-Figueras, J. Del Ser and S. Gil-López. "A Coral Reefs Optimization Algorithm for Optimal Mobile Network Deployment with Electromagnetic Pollution Control Criterion". *Applied Soft-Computing*, **submitted August 2012. JRC: 2,140**
4. A. M. Pérez-Bellido, S. Salcedo-Sanz, A. Portilla-Figueras, E. G. Ortiz-García and P. García-Díaz. "Evolutionary band-width allocation in reservation-based networks with Vickrey Auctions". *International Journal of Intelligent Information and Database Systems*. Vol. 3, pp. 75-81, 2009.

Conferencias nacionales e internacionales:

1. P. García-Díaz, S. Salcedo-Sanz, J. Plaza-Laina, A. Portilla-Figueras and J. Del Ser. "A discrete particle swarm optimization algorithm for mobile network deployment problems". *IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2012.
2. A. M. Pérez-Bellido, S. Salcedo-Sanz, J. A. Portilla-Figueras, E. G. Ortiz-García and P. García-Díaz. "An Agent System for Bandwidth Allocation in Reservation-based Networks using Evolutionary Computing and Vickrey Auctions". 1st KES-AMSTA (Symposium on Agent and Multi-Agent Systems) -Technologies and Applications, Lecture Notes in Artificial Intelligence. Vol. 4496, pp. 476-485. Wroclaw (Poland), 2007.
3. M. Pérez-Bellido, S. Salcedo-Sanz, J. A. Portilla-Figueras, E. G. Ortiz-García and P. García-Díaz. "A hybrid evolutionary programming algorithm for spread spectrum radar polyphase code design". *Genetic and Evolutionary Computation Conference (GECCO)*. Londres, UK. 2007.
4. A. Portilla-Figueras, S. Salcedo-Sanz, P. García-Díaz, K. Hackbarth. "A Genetic Algorithm For Solving The First Price Sealed Bid Auction In Communication Networks". *International Conference on Electronics, Hardware, Wireless and Optical Communications*. ISBN 960-8457-41-6, pp. 1-6. Alcalá de Henares (Madrid), 2006.
5. P. García-Díaz, K. Hackbarth, A. Portilla-Figueras, S. Salcedo-Sanz. "Simulation Of The Behavior Of Bidders In First-price Sealed-Bid Auctions On Communication Networks". *International Conference on Electronics, Hardware, Wireless and Optical Communications*. ISBN 960-8457-41-6, pp. 120-127. Alcalá de Henares (Madrid), 2006.
6. A. Portilla, S. Salcedo, S. Jiménez, P. García. "Cálculo Del Radio Celular En Sistemas WCDMA Mediante Programación Evolutiva". IV Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados. ISBN 84-9732-467-6, pp. 549-556. Granada, 2005.



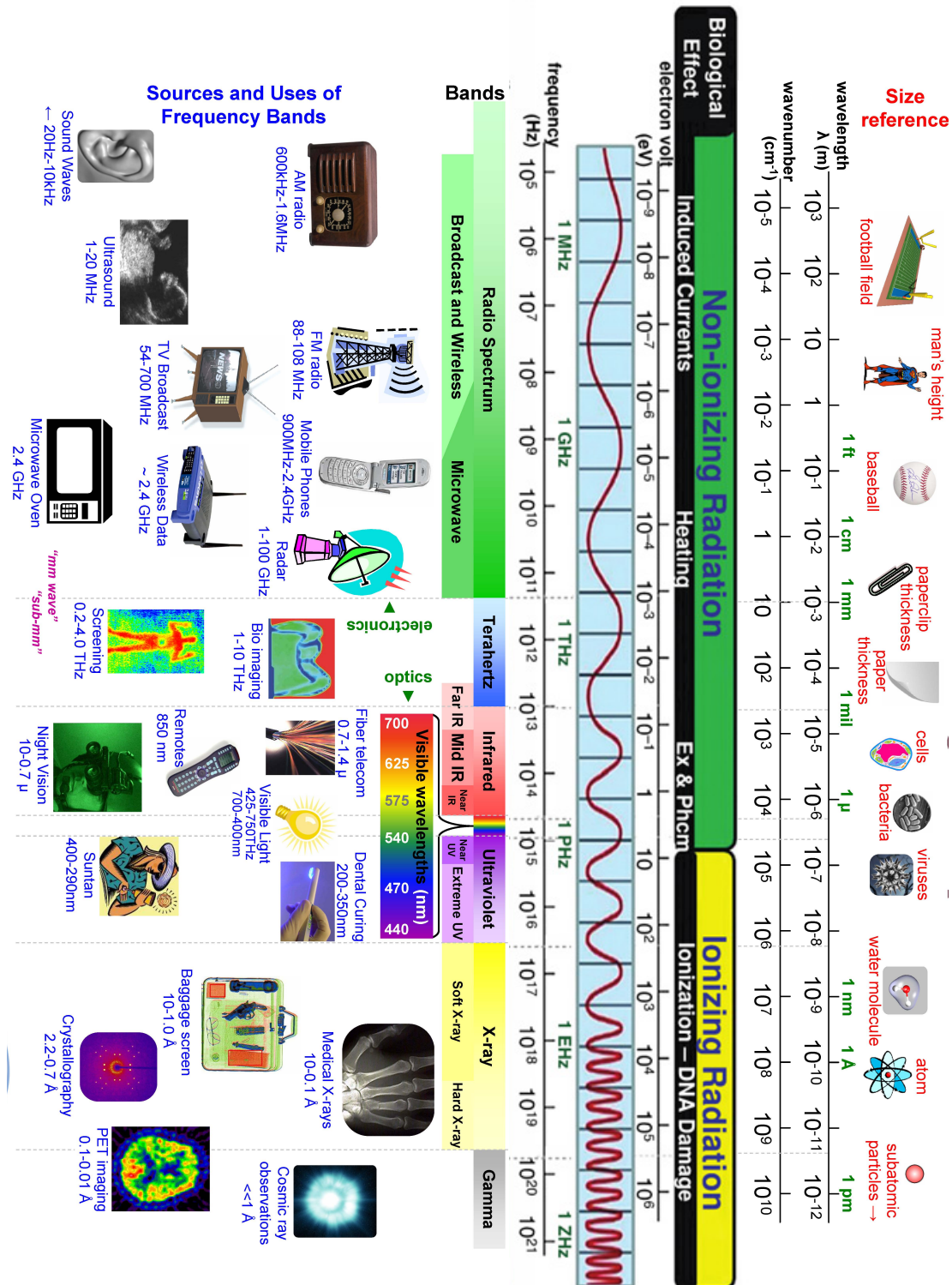
## Anexo A

# Tipos de radiación electromagnética

La radiación electromagnética es la combinación de campos eléctricos y magnéticos oscilantes que transportan energía a través del espacio o el vacío. Es la responsable de multitud de fenómenos físicos variados tales como la luz visible, el calor radiado, las ondas de radio y televisión. La radiación electromagnética se manifiesta de diversas formas, tales como rayos X, microondas o espectro visible. La Figura A.1 muestra el rango completo del espectro electromagnético. La parte inferior de la figura resume las aplicaciones tecnológicas del electromagnetismo mientras que la parte superior de la misma ofrece una clasificación de la radiación según el efecto que produce en la materia: radiación no-ionizante y radiación ionizante. La radiación no-ionizante es aquella que no posee energía suficiente para ionizar la materia o producir radiactividad en el cuerpo. Por contra, la radiación ionizante es la que sí puede generar radiactividad pudiendo extraer electrones del material que se radia.

El estudio de la radiación por el hombre es relativamente reciente y comenzó a explorar el espectro electromagnético por la parte superior del mismo. Tras el descubrimiento de los rayos X en el año 1895 por el físico Wilhelm Röntgen, Henri Becquerel comprobó la actividad de cristales de uranio le permitían revelar fotografías. El matrimonio Curie bautizó este fenómeno con el nombre de radiactividad y descubrieron nuevos materiales radiactivos como el torio, el polonio y el radio. En 1903 compartieron con Becquerel el Premio Nobel de Física por estos trabajos. En el siglo XX continuaron las investigaciones con Faraday, Maxwell y Rutheford principalmente. Progresivamente se encontraron aplicaciones abarcando todo el espectro de radio de la Figura A.1.

Como se ha dicho antes, la radiación ionizante es aquella en la que los fotones tienen la capacidad de ionizar la materia arrancando electrones de ésta. La energía de los fotones depende de la frecuencia de radiación, a frecuencias bajas su energía disminuye. Está científicamente comprobado que la radiación ionizante puede ser altamente nociva para la salud. Sin embargo, a día de hoy, no existe ningún estudio concluyente que indique que la radiación no-ionizante sea peligrosa para la salud. En esta categoría se incluyen la radiofrecuencia, las microondas, radiación de tipo *Terahertz* y los rayos infrarrojos. El trabajo de tesis tiene en cuenta exclusivamente el espectro radio, hasta los 300 GHz.



**Figura A.1:** Espectro de radiación electromagnética. Puede observarse que la radiación producida por las aplicaciones de radiocomunicación es de tipo no ionizante. Fuente: myscifi.org.



## Anexo B

# Algoritmo genético de agrupamiento aplicado al problema MNDP con restricción de capacidad

Describimos en este anexo algunos de los trabajos iniciados para la resolución del problema de optimización MNDP con restricción de capacidad, mediante un algoritmo genético de agrupamiento o *grouping genetic algorithm* (GGA).

La capacidad de una estación base se mide en Erlangs y define el tráfico máximo que puede gestionar. Cada una de las localizaciones asignadas a una determinada estación base consume una capacidad media dependiente del tipo de servicio demandado. De esta manera la estación base distribuye la capacidad disponible entre sus localizaciones hasta agotarla o mantener una capacidad sin asignar pero insuficiente para nuevas localizaciones.

El algoritmo GGA es un algoritmo genético modificado para adaptarlo a la resolución de problemas de grupos o *clustering* [37, 58]. Un problema clásico de *clustering* es el *Bin Packing Problem*, donde un total de elementos se distribuyen en el menor número de grupos o paquetes posible tal que el tamaño o volumen de los grupos no superen unas dimensiones de referencia. El problema MNDP con restricción de capacidad es también un problema de *clustering*, donde deseamos un despliegue con el mínimo número de estaciones base o grupos. La distribución de las localizaciones será tal que no se supere en ningún caso la capacidad máxima real de las estaciones base. De este modo surgen dos restricciones diferentes: la ya conocida restricción de cobertura, donde una estación base solo puede atender a localizaciones situadas en su área de cobertura; y la restricción de capacidad, donde no se puede asociar una localización a una estación base si la primera demanda una capacidad de tráfico superior a la disponible en la segunda.

### B.1 Planteamiento matemático del problema MNDP con restricción de capacidad

Al tener en cuenta que las estaciones base tienen una capacidad limitada, el problema formulado en el Capítulo 3 sufre algunas modificaciones importantes. La primera implicación es que el conjunto de tipos de estaciones  $\mathcal{T}$  se amplía al añadir la capacidad asignable de la estación base como un nuevo parámetro. Además ahora también permitimos que sean disjuntos los conjuntos  $\mathcal{O}$ ,  $\mathcal{P}$  y  $\mathcal{P}'$  de puntos o ubicaciones destacados en el área metropolitana  $A$ . De esta manera resulta más complicada la manipulación de datos en el algoritmo, porque debemos diferenciar los puntos que requieren demanda de servicio (conjunto  $\mathcal{P}'$ ) a los puntos o localizaciones donde es viable la instalación de equipos (conjunto  $\mathcal{P}$ ). A continuación generalizamos las nuevas especificaciones

para el problema MNDP:

- Conjunto de puntos de medida de campo eléctrico ( $V/m$ ) definido como  $\mathcal{O} = \{\mathbf{o}_i\}$ ,  $i \in \{1, \dots, O\}$ , donde cada punto  $\mathbf{o}_i$  tiene coordenadas  $(x_i^o, y_i^o) \in A$ .
- Conjunto de puntos  $\mathcal{P}$  donde es posible instalar estaciones base definido como  $\mathcal{P} = \{\mathbf{p}_i\}$ ,  $i \in \{1, \dots, P\}$ . Cada punto  $\mathbf{p}_i$  está queda determinado por sus coordenadas  $(x_i, y_i) \in A$  y su altitud medida en metros (m).
- Conjunto de puntos  $\mathcal{P}'$ , que recoge las localizaciones contenidas en  $A$  donde se desea ofrecer el servicio de telecomunicación, de forma que  $\mathcal{P}' = \{\mathbf{p}'_i\}$  con  $i \in \{1, \dots, M\}$ . Además, cada punto  $\mathbf{p}'_i$  tiene asociadas unas coordenadas  $(x'_i, y'_i)$  y una capacidad demandante (mE) a la red de telecomunicación.
- Consideramos también un conjunto de tipos de estaciones base definido como  $\mathcal{T} = \{\mathbf{t}_i\}$ ,  $i \in \{1, \dots, T\}$ . El tipo de una estación base depende del valor asignado en tres parámetros: frecuencia de trabajo de la estación, potencia de transmisión, capacidad asignable y coste económico.
- La función de coste  $\mathcal{C}(D)$  es la misma que definimos en el Capítulo 3, donde  $\mathcal{C}(D)$  indica el coste económico de las estaciones base que componen la red de telecomunicación, dependiendo del número de equipos y del tipo de los mismos.

El resultado óptimo del problema coincide también con lo expuesto en el Capítulo 3: deseamos obtener una solución de despliegue de red tal que se minimicen los valores de radiación electromagnética en la zona  $A$ , del coste económico total en la instalación de estaciones base y del grado de no cobertura sobre el conjunto objetivo  $\mathcal{P}'$ . El grado de cobertura será total cuando todas las localizaciones de  $\mathcal{P}'$  estén asignadas a estaciones base y además éstas dispongan de capacidad suficiente para satisfacer toda demanda.

La función de evaluación o función de *fitness* coincide con la indicada en el Capítulo 3, que replicamos en la Ecuación B.1, donde lo único que varía es la forma de calcular el término  $P_{no\ cober}$ . Dicho término calcula el porcentaje de localizaciones cuya demanda de servicio no es satisfecha por ninguna estación base, bien por falta de cobertura o bien por falta de capacidad en estaciones próximas.

$$\mathcal{F} = k_1 \cdot \Delta E_T + k_2 \cdot \mathcal{C}_{\mathcal{T}} + k_3 \cdot P_{no\ cober} \quad (\text{B.1})$$

Dado que el conjunto  $\mathcal{P}'$  no abarca en general toda la superficie de  $A$ , podemos reducir este área para calcular con mayor exactitud el número de estaciones base mínimo necesarias. Definimos para ello un conjunto de zonas  $\mathcal{Q}$  con geometría de paralelogramo que incluyan todos los elementos de  $\mathcal{P}'$ . Formalmente expresamos este conjunto como  $\mathcal{Q} = (\mathbf{z}_i), i \in \{1, \dots, Q\}$ . La definición de  $\mathcal{Q}$  no es única, la razón de su existencia es reducir la superficie de terreno a cubrir con las necesarias estaciones base. De forma análoga podemos definir también otro conjunto de zonas que delimiten las localizaciones del conjunto  $\mathcal{P}$  donde es factible la instalación de las estaciones base.

## B.2 Algoritmo genético de agrupamiento desarrollado para el problema MNDP

Algunas de las funciones o procesos descritos para el EA en el Capítulo 4 no sufren variación alguna en el GGA, como es el caso de la función de evaluación analizado en el párrafo anterior.

También es el caso del operador de selección de individuos y de las condiciones de parada del algoritmo. Sin embargo, otros aspectos resultan notablemente alterados, entre ellos destaca la codificación del problema. Y como consecuencia de ello, se modifican los operadores de exploración del algoritmo, es decir, los operadores de cruce y mutación. En esta sección vamos a describir la codificación del problema, la generación de la población inicial, los operadores de cruce y mutación, la operación de reparación y el proceso de búsqueda local para el algoritmo GGA.

### B.2.1 Codificación del problema

La codificación del problema en un algoritmo de agrupamiento está basada en individuos de longitud variable donde se diferencian claramente dos partes [37]: la parte de asignación o *assignment part* y la parte de agrupamiento o *grouping part*. En el caso del problema MNDP vamos a añadir una tercera. De este modo, cada individuo está formado por un conjunto de  $N$  estaciones base y codificado en las tres secciones siguientes:

$$c = \{(a_i \mid p_j \mid t_j)\}, \quad a_i \in \mathcal{P}, \quad i = 1, \dots, M; \quad p_j \in \mathcal{P}, \quad t_j \in \mathcal{T}, \quad j = 1, \dots, N$$

La primera parte  $a_i$  es la *assignment part* propiamente dicha, indica la estación base a la que se asigna cada localización demandante del servicio. La *assignment part* es de longitud fija  $M$ , siendo  $M$  el cardinal del conjunto de localizaciones  $\mathcal{P}'$  donde se quiere ofrecer el servicio de telecomunicación. Para cada elemento  $a_i$ , su posición en la solución se hace corresponder con la posición de una localización en  $\mathcal{P}'$ . Por ello es importante mantener siempre ordenados los elementos de  $\mathcal{P}'$ . El contenido de  $a_i$  es siempre un elemento del conjunto  $\mathcal{P}$ , indicando así a qué estación base se encuentra asignada la localización correspondiente.

La segunda parte  $p_j$  la denominamos *group part* y contiene tantos elementos ( $N$ ) como estaciones base tiene el individuo o solución. El contenido de  $p_j$  es el conjunto de identificadores de las localizaciones donde se ubican dichas estaciones en  $\mathcal{P}$ . Por último,  $t_j$  define la *typing part*, de la misma longitud que  $p_j$ , donde se registra el tipo de estación base y en el mismo orden que en  $p_j$ . Es importante destacar que de esta forma la solución se codifica con una sucesión ordenada de  $(M+N+N)$  elementos de los conjuntos  $\mathcal{P}$  y  $\mathcal{T}$ .

El individuo codificado tiene longitud variable puesto que el número  $N$  de estaciones también es variable. Para lograr una codificación única de la solución, mantenemos ordenadas las localizaciones de  $p_j$ , de manera similar a como explicamos para el EA. La Figura B.1 muestra un ejemplo de codificación para una sencilla solución compuesta por solo cuatro estaciones base, localizadas en las ubicaciones 2, 3, 4 y 6 del conjunto  $\mathcal{P}$  y con tipos 1, 3, 3 y 1 respectivamente. El conjunto de puntos a los que se desea dar servicio  $\mathcal{P}'$  tiene un tamaño de  $M = 14$  puntos, esta será la longitud de la sección  $a_i$ . Esta sección establece la asignación de cada uno de estos puntos a una de las cuatro estaciones base. Cuando la localización queda sin satisfacer su demanda, se registra el valor “0” en la posición correspondiente de  $a_i$ .

Según la Figura B.1 la solución deja cuatro ubicaciones sin cobertura, tres asignadas a la estación base instalada en la localización “2”, tres puntos asociados a la estación base ubicada en “3”, dos puntos asociados a la estación instalada en la localización “4” y otros dos para la estación base colocada en la ubicación con identificador “6”.

### B.2.2 Generación de la población inicial

La primera generación del algoritmo GGA está constituida por una población de  $N_{ind}$  individuos completamente aleatorios de forma análoga al EA descrito en la Sección 4.1.2. Sin embargo,

0	2	2	4	3	4	0	0	6	3	3	6	0	2	2	3	4	6	1	3	3	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

**Figura B.1:** Ejemplo de codificación de una solución para el problema MNDP con el algoritmo GGA.

debido a la restricción de capacidad de las estaciones base, se modifica la manera de realizar la estimación del número de estaciones necesarias para satisfacer toda la demanda total. Como para el EA, empleamos el intervalo  $[N_{MinB}, N_{MaxB}]$ , siendo éstos los límites inferior y superior para el rango de número de estaciones base de un individuo de creación aleatoria.

Para cada zona  $z_i$  de  $\mathcal{Q}$  vamos a calcular el valor  $B_i$ , definiendo  $B_i$  como una sobre-estimación del número mínimo de estaciones del tipo más desfavorable que deberían cubrir la zona  $z_i$  para satisfacer las restricciones de demanda y cobertura en dicha zona.  $B_i$  se determina según la Ecuación B.2, donde  $B_{i,cap}$  y  $B_{i,cov}$  son las estimaciones de  $B_i$  atendiendo a las restricciones de capacidad y cobertura por separado. El cálculo de  $B_{i,cov}$  coincide con lo expuesto en la Sección 4.1.2.

$$B_i = \max\{B_{i,cap}, B_{i,cov}\} \quad (B.2)$$

A partir del cálculo de  $B_i$  para cada zona, la estimación de  $B_{max}$  se toma como la suma de las estimaciones  $B_i$  para cada una de las zonas, tal y como muestra la Ecuación B.3. Finalmente determinamos los límites del intervalo  $[N_{MinB}, N_{MaxB}]$  mediante las Ecuaciones B.4 y B.5.

$$B_{max} = \sum_{i=1}^Q B_i \quad (B.3)$$

$$N_{MinB} = \lceil 0, 2B_{max} \rceil \quad (B.4)$$

$$N_{MaxB} = \lceil 0, 4B_{max} \rceil \quad (B.5)$$

El pseudo-código para este algoritmo se recoge en el Algoritmo 17. Tenemos que destacar que es necesario verificar la viabilidad de la solución mediante la operación de reparación del individuo, ya que se realiza una asociación aleatoria de localizaciones a estaciones base en la penúltima sentencia del pseudo-código.

### B.2.3 Operador de cruce o recombinación

El operador de cruce que hemos utilizado en el algoritmo GGA está basado en [37] y presta atención a la longitud variable de los individuos a la vez que consigue que los individuos descendientes hereden fragmentos completos de información de ambos progenitores. Para generar un individuo a partir de otros dos elegidos al azar, el procedimiento general consiste en realizar una copia de uno cualquiera de ellos y sobre ésta insertar un fragmento de la parte de agrupamiento o *grouping part* del otro progenitor con las correspondientes asignaciones en su *assignment part*. Las Figuras B.2 y B.3 representan un sencillo ejemplo de cruce de dos progenitores para generar un descendiente. Ambas son el mismo ejemplo, mientras que la Figura B.2 muestra el cruce desde el punto de vista de la codificación de las soluciones en individuos, la Figura B.3 lo hace a través de la representación gráfica de las soluciones.

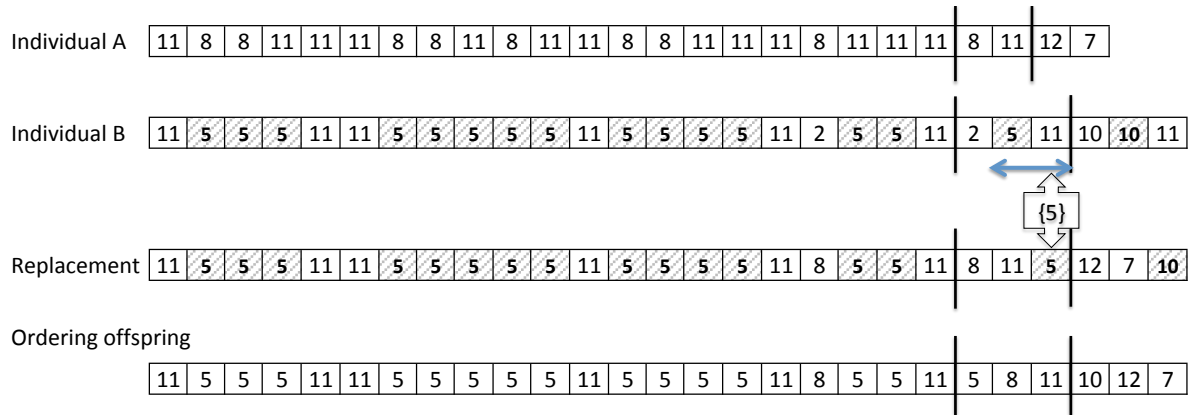
**Algorithm 17** Pseudo-código para la generación de la población inicial en el GGA**Require:** Number  $N_{ind}$  of individuals and range  $[N_{MinB}, N_{MaxB}]$ **Ensure:** Feasible random population

```

1: for each individual  $i$  of  $N_{ind}$  do
2:   Generate a random number of BTSs  $N_i \in [N_{MaxB}, N_{MinB}]$ 
3:   for each BTS of  $N_i$  do
4:     Select a random location for the BTS from  $\mathcal{P}$ 
5:     Select a random type for the BTS from  $\mathcal{T}$ 
6:   end for
7:   Ordering locations in ascending order.
8:   Generate random associations to BTSs.
9:   Repairing process
10: end for

```

En la Figura B.2 se representa cada individuo como un vector de longitud variable en la que se han delimitado las partes de asignación  $a_i$ , agrupamiento  $p_j$  y tipo  $t_j$ , según se describió en la Sección B.2.1. Los individuos progenitores se han denominado *Individual A* e *Individual B*. El proceso se inicia con una réplica de *Individual A*. A continuación se toma un fragmento aleatorio de la parte de agrupamiento *group part* de *Individual B*, que en este caso resulta ser el subconjunto de localizaciones para estaciones base  $\{5, 11\}$ . De este subconjunto se transmiten únicamente las localizaciones no presentes en *Individual A*. De modo que solo se añade al nuevo individuo la localización número “5” y su correspondiente tipo “10”. En la figura se han marcado estos elementos con un ligero entramado en color gris. En la sección *assignment part* del nuevo individuo se modifican todas las asignaciones que en *Individual B* correspondieran a la localización “5”. Estos elementos también aparecen destacados en la figura en el individuo etiquetado con “*Replacement*”. Por último se deben ordenar las localizaciones de las partes de agrupamiento  $b_j$  y  $t_j$ .

**Figura B.2:** Ejemplo de cruce de dos individuos en el algoritmo GGA para el problema MNDP.

Podemos ver la representación gráfica de los tres individuos de este sencillo ejemplo en la Figura B.3. En ella se destacan las localizaciones que demandan servicio (conjunto  $\mathcal{P}'$ ) mediante los puntos distribuidos por el cuadrante formado por los ejes X e Y. Las zonas viables para la instalación de estaciones base (conjunto  $\mathcal{P}$ ) se representan por paralelogramos. Para simplificar el ejemplo hemos incluido una única zona para cada uno de los conjuntos  $\mathcal{P}'$  y  $\mathcal{P}$ . La zona teórica de cobertura de cada estación base se delimita mediante una circunferencia. Cada una de las

estaciones base de la solución se representa con un color diferente. La ubicación de una estación base se destaca con un punto “estrella” del mismo color que su circunferencia. Las localizaciones asignadas a una estación base se representan por puntos con el mismo color que la circunferencia que delimita su zona de influencia. De este modo, el código de colores nos ayuda a distinguir en la figura asignaciones a diferentes estaciones. Sin embargo los colores de una solución y otra son completamente independientes, ya que el algoritmo los asigna según el orden en los que los va representando en la gráfica. Este orden coincide con el orden en la parte  $b_j$  y es el siguiente: azul oscuro, verde y azul claro.

En la Figura B.3(a) podemos reconocer las dos estaciones base que componen el *Individual A*, cuyos identificadores son  $\{8,11\}$ . Son de tipos diferentes y puede verse que los radios de ambas circunferencias son también muy distintos (se corresponden con los radios de cobertura de ambas estaciones). La Figura B.3(b) es la representación del *Individual B*, con tres estaciones base. Comparando ambas figuras podemos encontrar dos circunferencias centradas en el mismo punto. Este punto se corresponde con la localización “11”. Sin embargo las circunferencias tienen radios de diferente longitud porque los tipos de las estaciones base son también distintos, “7” en *Individual A* y representada en color verde, mientras que en el *Individual B* el tipo es “11” y la circunferencia es de color azul claro. En la Figura B.3(c) muestra por su parte las tres estaciones base que forman el individuo descendiente *offspring*, puede verse que contiene todas las estaciones base del *Individual A*, aunque las representa con colores distintos debido al orden en el que figuran en la codificación de la Figura B.2, y una sola del *Individual B*, que se corresponde con el color verde en la Figura B.3(b) y el color azul oscuro en el nuevo individuo en la Figura B.3(c). El siguiente paso para completar el proceso sería la reparación de este nuevo individuo, no incluido en la figura.

El Algoritmo 18 resume el pseudo-código de la operación de cruce o recombinación. El proceso siempre debe finalizar con la reparación del individuo descendiente *offspring*, puesto que es posible que en la asignación a las nuevas localizaciones se haya violado la restricción de capacidad de alguna de las estaciones base insertadas.

---

**Algorithm 18** Pseudo-código del operador de cruce en el GGA

---

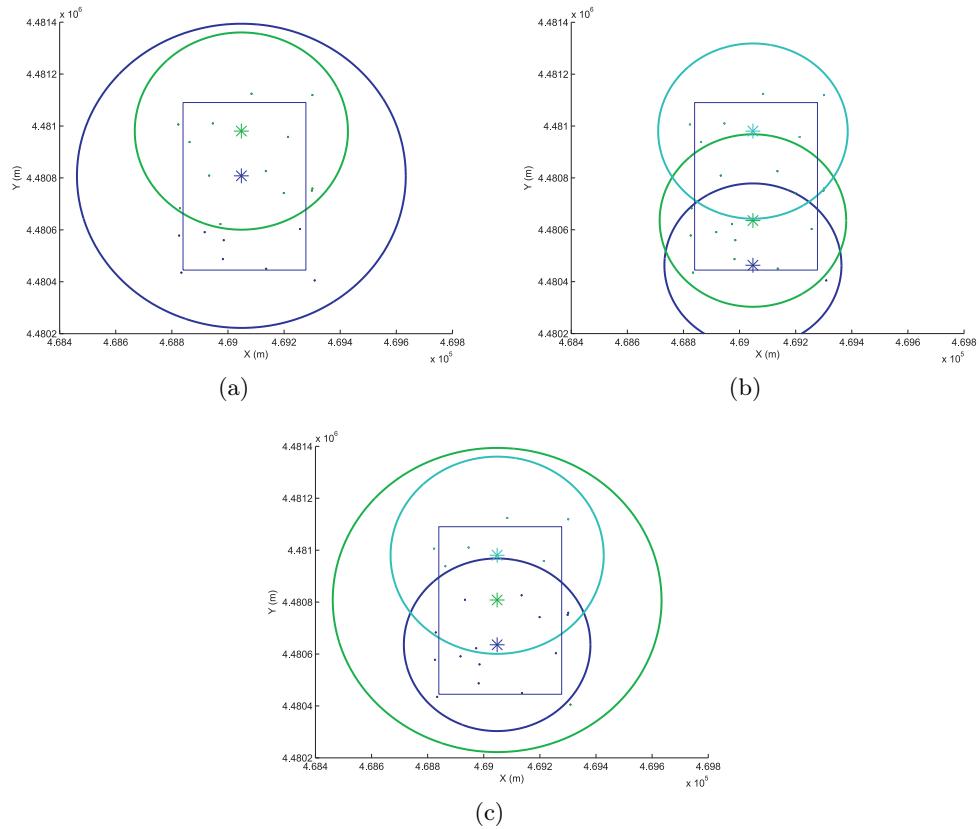
**Require:** Random couple: individual A and individual B

**Ensure:** Single offspring: feasible individual

- 1: Copy individual A in offspring
  - 2: Select random range of BTSs in individual B
  - 3: Let  $C = \{\text{BTSs of the range no present in offspring}\}$
  - 4: Insert C in offspring
  - 5: Copy all assignments of BTSs in set C from individual B to offspring
  - 6: Sort BTSs offspring in ascending order
  - 7: Repairing offspring
  - 8: Local searching offspring if required
- 

#### B.2.4 Operador de mutación

El proceso de mutación de un individuo se realiza exclusivamente en las partes de agrupamiento  $b_j$  y  $t_j$ . Para ello se genera una máscara binaria y aleatoria de longitud  $N+N$ , coincidiendo con la longitud de las dos partes referidas. Como en el EA, los “1” de la máscara indican que genes han de ser mutados, eligiendo valores aleatorios en los conjuntos  $\mathcal{P}$  y  $\mathcal{T}$ , según corresponda. De esta forma lo que varía es la ubicación y/o el tipo de las estaciones base de la solución. El Algoritmo 19 ofrece las instrucciones descritas para la operación de mutación. Es un requisito



**Figura B.3:** Ejemplo simplificado de la función del operador de cruce o recombinación en el algoritmo GGA. Representación gráfica de las soluciones de la Figura B.2, donde a partir de dos individuos aleatorios B.3(a) y B.3(b) se genera un nuevo individuo *offspring* B.3(c).

imprescindible reparar el individuo resultante, puesto que con el cambio de ubicación de las estaciones base es muy probable dejar de satisfacer la restricción de cobertura.

---

**Algorithm 19** Pseudo-código del operador de mutación en el GGA

---

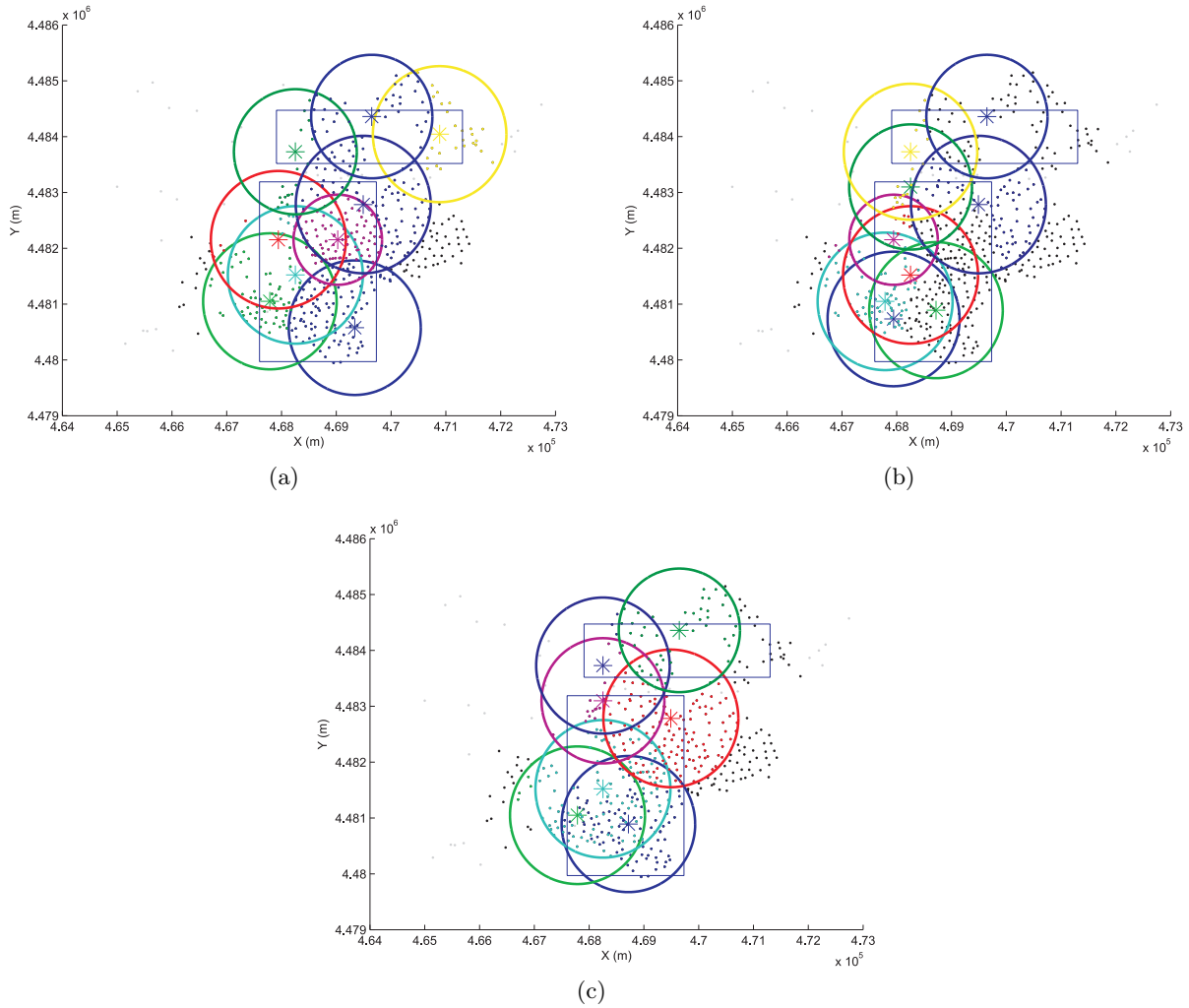
**Require:** An individual

**Ensure:** Feasible mutated individual

- 1: Generate random binary mask L and mask T of length N (number of BTSs)
  - 2: In group part, change locations to other random ones where mask L has bit '1'
  - 3: In typing part, change types to other random ones where mask T has bit '1'
  - 4: Ordering stations
  - 5: Remove stations with repeated location
  - 6: In assignment part, delete assignments to stations no present any more
  - 7: Repairing and local search of individual
- 

La Figura B.4 muestra un ejemplo de mutación para un individuo con  $N=9$  estaciones base. La Figura B.4(a) representa gráficamente este individuo. Puede observarse que aparecen dos rectángulos que delimitan la zona donde deben ubicarse todas las estaciones base. De hecho los centros de todas las circunferencias se encuentran dentro de dichos rectángulos. La Figura B.4(b) es la representación del individuo mutado, con una máscara  $L = \{0,1,1,1,0,1,1,0,0\}$  para la parte *group part*. El código de colores utiliza el orden creciente en las localizaciones, de modo que los

colores entre dos individuos distintos no tienen correspondencia. Finalmente, la Figura B.4(c) ofrece la representación gráfica del individuo mutado tras los procesos de reparación y búsqueda local. La reparación es necesaria, con ella se verifica el cumplimiento de las restricciones de cobertura y capacidad. La operación de búsqueda local permite encontrar soluciones similares con mejor valor de *fitness*. Con ambas operaciones, la solución ha reducido su número de estaciones base a 7, puesto que había dos en la Figura B.4(b) que resultaban prescindibles (representadas con colores azul oscuro y rosa en dicha figura).

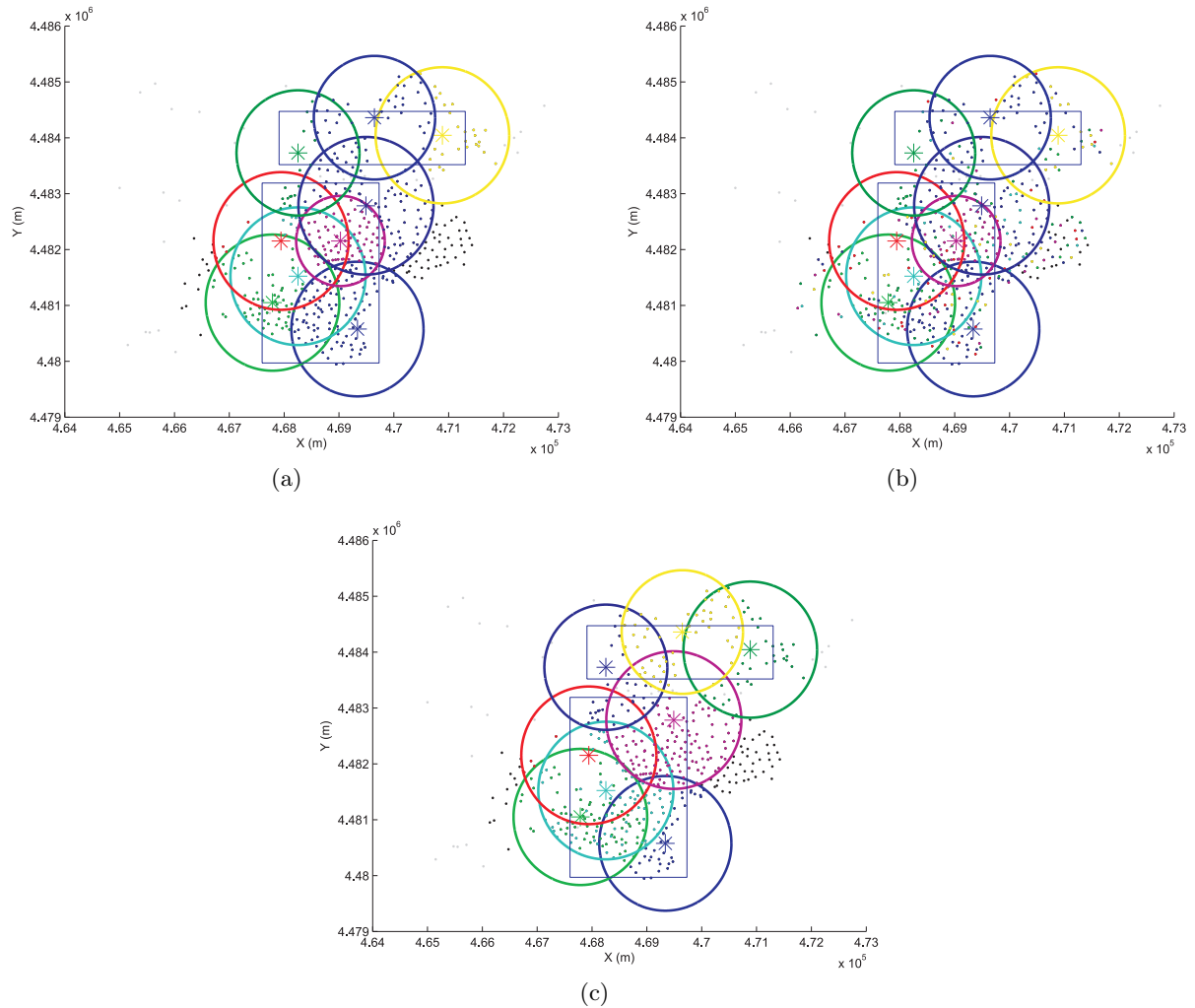


**Figura B.4:** Ejemplo de la operación de mutación sobre una solución del problema MNDP con el algoritmo GGA. La solución previa a la mutación, formada por nueve estaciones base, se representa en la Figura B.4(a). El resultado de la mutación es la solución de la Figura B.4(b). Tras las operaciones de reparación y búsqueda local, se obtiene la solución de la Figura B.4(c), constituida por siete estaciones.

En la Figura B.5 mostramos un ejemplo de como la mutación con máscara binaria sobre la parte de asignación no resulta eficaz, puesto que la solución resultante se extremadamente similar a la solución de partida. Esto es causado porque las alteraciones que produce la máscara sobre la parte de asignación son en su mayoría destruidas por el efecto de la reparación en el cumplimiento de la restricción de cobertura. De este modo las Figuras B.5(a) y B.5(c) representan la misma



solución aunque las circunferencias de una y otra tengan distintos colores.



**Figura B.5:** Ejemplo de mutación no eficaz sobre la parte de asignación del individuo. La Figura B.5(a) muestra una solución previa a la mutación. La Figura B.5(b) es el resultado inmediato de la mutación. Después de verificar las restricciones de cobertura y capacidad se obtiene la solución de la Figura B.5(c), que es notablemente parecida a la solución de la Figura B.5(a).

### B.2.5 Proceso de reparación de un individuo

En los apartados anteriores hemos descrito operaciones de exploración sobre los individuos, que pueden modificar de tal manera su composición que dejen de corresponder a soluciones viables del problema MNDP. Esto ocurrirá siempre que las modificaciones violen las restricciones de cobertura o capacidad impuestas:

- Una estación base no puede atender localizaciones que se encuentren fuera de su zona de cobertura o influencia.
- La suma de las capacidades demandadas de todas las localizaciones asignadas a una estación base no puede superar la capacidad total de dicha estación.

En el Algoritmo 20 se propone la secuencia de operaciones para reparar los individuos que no se identifiquen con soluciones viables. En primer lugar se comprueba la restricción de cobertura, asignando todas las localizaciones del área  $A$  a alguna estación base que la pueda cubrir, salvo que la localización se encuentre fuera de cualquier zona de cobertura. La asignación de una localización a una estación base es siempre aleatoria entre las estaciones que la alcanzan.

En segundo lugar se examinan todas las estaciones en orden aleatorio. Para cada una de ellas se verifica que la capacidad demandada no supera la disponible. Si esto sucediera se seleccionan al azar localizaciones candidatas a ser eliminadas hasta el cumplimiento de la restricción de capacidad. Las localizaciones candidatas se intentan asignar a nuevas estaciones que dispongan de capacidad suficiente, mientras que si esto no es posible se les permite mantener la asignación a la estación base que se está examinando con cierta probabilidad (eligiendo una nueva localización candidata) o se suprime dicha asignación con probabilidad complementaria. El proceso se repite hasta el cumplimiento de la restricción de capacidad en la estación base analizada.

Finalmente, no se aceptan en la solución estaciones base con menos asignaciones que un umbral definido como  $\beta\%$ . Nótese que el resultado del proceso de reparación de un individuo no es único: debido al orden aleatorio en el que se revisan tanto las estaciones como las localizaciones, el algoritmo podría obtener diferentes resultados en simulaciones distintas. Este hecho no supone, en principio, ningún problema para la convergencia del algoritmo.

---

**Algorithm 20** Pseudo-código de la operación de reparación del individuo en el GGA

---

**Require:** An individual

**Ensure:** Feasible individual

```

1: Generate the  $N \times M$  coverage matrix
2: for all demand location do
3:   if location no assigned OR assigned without coverage then
4:     Assign to a random BTS with coverage
5:   end if
6: end for
7: for each station in random order do
8:   if Not fulfill capacity constrain then
9:     Release asignations
10:  end if
11: end for
12: for each station of the individual do
13:   if less asignations than  $\beta\%$  then
14:     Remove station
15:   end if
16: end for

```

---

### B.2.6 Proceso de búsqueda local

El objetivo de la operación de búsqueda local es mejorar el valor de *fitness* de un individuo, realizando algún tipo de modificación estratégica que aproxime la solución la óptima global. Existen diferentes criterios para llevar a cabo esta tarea. En el EA nos centramos en el desplazamiento o eliminación de estaciones prescindibles, es decir, aquellas que aportan poco o nada a la calidad de la solución. En el GGA tratamos, por un lado, que ninguna localización quede sin servicio si existiendo alguna estación base cercana con capacidad disponible. Por otro lado, buscamos también estaciones prescindibles, liberando antes sus asignaciones y asociándolas a otras estaciones respetando siempre las restricciones de cobertura y capacidad. De esta manera

intentamos recolocar las asignaciones de forma más eficiente, mejorando sin duda el valor de la función objetivo.

El Algoritmo 21 ofrece el pseudo-código resumido del proceso de búsqueda local en un individuo. En la línea 8 del mismo establecemos un orden aleatorio para la revisión y búsqueda de estaciones base prescindibles en el 90 % de los casos, mientras que en el 10% restante, la revisión comienza por las estaciones con menor número de asignaciones. En el ejemplo de la Figura B.3, varias estaciones pueden resultar prescindibles si no se viola en ningún caso la restricción de capacidad en las de mayor área de cobertura. Esta figura está incluida en la Sección B.2.3, donde no se aplicó la operación de búsqueda local intencionadamente para prestar plena atención al operador de cruce. Las asignaciones restablecidas en el proceso de revisión de estaciones (bucle *for* entre las líneas 8 y 16 del pseudo-código) son siempre etiquetadas para no deshacerlas posteriormente dentro del bucle.

Finalmente, como en la reparación de individuos, solo se aceptan estaciones base que tengan un mínimo de  $\beta$  % de localizaciones asignadas sobre el total. El resultado final es un individuo similar al inicial pero con mejor valor de *fitness*, como es el caso de las Figuras B.4(b) y B.4(c), en el que se elimina una estación base, reduciendo por tanto el el incremento de campo eléctrico en  $A$ , es coste total de la red, sin aumentar con ello el porcentaje de demanda no satisfecha.

---

**Algorithm 21** Pseudo-código del procedimiento de búsqueda local en un individuo en el GGA

---

**Require:** A feasible individual

**Ensure:** A similar and feasible individual with better fitness

```

1: Generate  $N \times M$  coverage matrix
2: for each no assinged location in random ordering do
3:   if there is any stations with coverage and capacity enough then
4:     Select a random station of the list
5:     Change the assignation to new station
6:   end if
7: end for
8: for each station of the individual in random order (90%) or ascending order of number of
   allocations (10%) do
9:   for all locations assigned to the station do
10:    if there is/are other/s station/s with coverage and capacity enough then
11:      Select a random station of the list
12:      Change the assignation to new station
13:      Check the action to avoid release this assignation later
14:    end if
15:  end for
16: end for
17: for each station of the individual do
18:   if less asignations than  $\beta\%$  then
19:     Remove station
20:   end if
21: end for

```

---



# Bibliografía

- [1] B. Abolhassani, J. E. Salt, D. Dodds. A two-phase genetic k-means algorithm for placement of radioports in cellular networks. *IEEE Trans. Syst. Man Cybern. B Vol. 34(1)*, pages 533–538, 2004.
- [2] F. Afshinmanesh, A. Marandi, and A. Rahimi Kian. A novel binary Particle Swarm Optimization method using artificial immune system. In *Proceedings IEEE International Conference on Computer Tool (EUROCON)*, pages 217–220, 2005.
- [3] L. Araujo and C. Cervigón. Algoritmos evolutivos. Un enfoque práctico. *Ra-Ma*, ISBN 978-84-7897-911-0, pages 27–46, 2009.
- [4] Asociación Española Contra el Cáncer (AECC). Campos electromagnéticos y salud: información y percepción de riesgo en la población general. Estudio realizado por demoscopia para la AECC. Abril, 2004. Enlace consultado en junio 2011.
- [5] R. Ates. Aggressive behaviour in corals. *Freshwater and Marine Aquarium*, vol. 12(8), pages 104–112, 1989.
- [6] R. D. Barnes. Invertebrate zoology. *Orlando, FL, USA: Harcourt Brace Jovanovich, Inc., Ed. 5*, pages 149–163, 1987.
- [7] L. B. Booker. *Intelligent behavior as an adaptation to the task environment*. PhD thesis, Logic of Computers Group, University of Michigan, Ann Arbor, Michigan, 1982.
- [8] D. Bratton and J. Kennedy. Defining a standard for Particle Swarm Optimization. *IEEE Swarm Intelligence Symposium*, pages 120–127, 2007.
- [9] D. A. Brazeau, D. F. Gleason, and M. E. Morgan. Self-fertilization in brooding hermaphroditic caribbean corals: evidence from molecular markers. *Journal of Experimental Marine Biology and Ecology*, Vol. 231(2), pages 225–238, 1998.
- [10] A. Brindle. *Genetic algorithms for function optimization*. PhD thesis, Department of Computer Science, University of Alberta, Edmonton, Alberta, 1981.
- [11] D. E. Burkepille and M. E. Hay. *Coral reefs*. Encyclopedia of Ecology, 2008.
- [12] A. Carlisle and G. Dozier. An Off-the-shelf PSO. In *Proceedings of the Particle Swarm Optimization Workshop*, Vol. 1, pages 1–6, 2001.
- [13] N. E. Chadwick. Interspecific aggressive behavior of the Corallimorpharian *Corynactis Californica* (Cnidaria: Anthozoa): effects on sympatric corals and sea anemones. *Biological Bulletin*, Vol. 173, pages 110–125, 1987.

- [14] S. Chamberland. An efficient heuristic for the expansion problem of cellular wireless networks. *Comput. Oper. Res. Vol. 31*, pages 1769–1791, 2004.
- [15] S. Chandrasekaran, S. G. Ponnambalam, R. K. Suresh, and N. Vijayakumar. A hybrid discrete Particle Swarm Optimization algorithm to solve flow shop scheduling problems. In *Proc. Int. Conf. Cybern. Intell. Syst*, pages 1–6, 2006.
- [16] C.K. Chou, H. Bassen, J. Osepchuk, Q. Balzano, R. Petersen, M. Meltz, R. Cleveland, JC. Lin, and L. Heynick. Radio frequency electromagnetic exposure: Tutorial review on experimental dosimetry. *Bioelectromagnetics 17*, pages 195–208, 1996.
- [17] C. H. Chu, G. Premkumar, H. Chou. Digital data networks design using genetic algorithms. *Eur. J. Oper. Res. Vol. 127*, pages 140–158, 2000.
- [18] M. Clerc. The swarm and the queen: towards a deterministic and adaptive Particle Swarm Optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation (CEC)*, pages 1951–1957, 1999.
- [19] M. Clerc. Discrete Particle Swarm Optimization. New York, 2004. New Optimization Techniques in Engineering. Springer-Verlag.
- [20] M. Clerc and J. Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation, Vol. 6(1)*, pages 58–79, 2002.
- [21] C. A. Coello Coello. *Introducción a la computación evolutiva (Notas de curso)*. Av. Instituto Politécnico Nacional No. 2508. Col. San Pedro Zacatenco. Méjico. D. F. 07300, Departamento de Ingeniería Eléctrica. Sección de Computación, 2004.
- [22] Colegio Oficial de Ingenieros de Telecomunicación. *Emisiones radioeléctricas: normativa, técnicas de medida y protocolos de certificación*. ISBN 978-84-935049-7-7, 2008.
- [23] Comisión del Mercado de las Telecomunicaciones. Informe anual 2010.
- [24] Comisión del Mercado de las Telecomunicaciones. Informe económico sectorial. 2011.
- [25] P. Cortés, J. Larrañeta, L. Onieva, J. M. García, M. S. Caraballo. Genetic algorithm for planning cable telecommunication networks. *Appl. Soft Comput. Vol. 1(1)*, pages 21–33, 2001.
- [26] C. Cruz Corona. *Estrategias cooperativas multiagentes basadas en Soft Computing para la solución de problemas de optimización*. PhD thesis, Universidad de Granada, 2005.
- [27] L. D. Davis. *Handbook of genetic algorithms*. Van Nostrand Reinhold Company, New York, 1991.
- [28] A. K. de Jong. *An analysis of the behaviour of a class of genetic adaptive systems*. PhD thesis, University of Michigan, 1975.
- [29] A. Diéguez Alberte, R. Ovejero Málaga, and A. Robledo Ibáñez. *Implementación de juegos usando algoritmos evolutivos*. Facultad de Informática, Dpto. Sistemas Informáticos y programación, Universidad Complutense de Madrid, 2003.

- [30] M. Dorigo. *Optimization, learning and natural algorithms, Ph.D. Thesis*. Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.
- [31] M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: optimization by a colony of cooperating agents. In *IEEE Transactions on Systems, Man and Cybernetics, part B: Cybernetics, Vol. 26(1)*, pages 29–41, 1996.
- [32] R. C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan. Piscataway, NJ: IEEE Service Center*, pages 39–43, 1995.
- [33] R. C. Eberhart and Y. Shi. Comparing inertia weights and constriction factors in Particle Swarm Optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation (CEC), Vol. 1*, pages 84–88, 2000.
- [34] R. C. Eberhart and Y. Shi. Particle Swarm Optimization: developments, applications and resources. In *Proceedings Congress on Evolutionary Computation, Seoul, Korea, Piscataway, NJ: IEEE Service Center*, 2001.
- [35] A. E. Eiben and J. E. Smith. *Introduction to evolutionary computing*. Springer-Verlag, Natural Computing Series, Edición 1, 2003.
- [36] Euro Cost 231. *Urban transmission loss models for mobile radio in the 900 and 1800 MHz bands*. European Cooperation in the Field of Scientific and Technical Research EURO-COST 231, Revision 2, The Hague, 1991.
- [37] E. Falkenauer. *Genetic algorithms for grouping problems*. New York, Wiley, 1998.
- [38] D. B. Fogel. Evolutionary computation: toward a new philosophy of machine intelligence. In *IEEE Press Series on Computational Intelligence*, 2006.
- [39] R. Forsati, A. T. Haghighat, and M. Mahdavi. Harmony Search based algorithm for bandwidth-delay-constrained least-cost multicast routing. In *Computer Communications, Vol. 31*, pages 2505–2519, 2008.
- [40] R. Forsati and M. Mahdavi. Web text mining using Harmony Search. In *Recent advances in Harmony Search Algorithm*, pages 51–64, 2010.
- [41] Y. Fukuyama and H. A. Yoshida. A Particle Swarm Optimization for reactive power and voltage controlling electric power systems. In *Proceedings of IEEE International Congress on Evolutionary Computation. Seoul, Korea*, pages 496–502, 2001.
- [42] Z. W. Geem. Improved Harmony Search from ensemble of music players. In *Knowledge-Based Intelligent Information and Engineering Systems, part I, Vol. 4251*, pages 86–93, 2006.
- [43] Z. W. Geem. Optimal cost design of water distribution networks using Harmony Search. In *Engineering Optimization, Vol. 38(3)*, pages 259–280, 2006.
- [44] Z. W. Geem. Harmony Search algorithm for solving Sudoku. In *Lecture Notes on Computer Science, Vol. 4692*, pages 371–378, 2007.
- [45] Z. W. Geem, J. H. Kim, and G. V. Loganathan. A new heuristic optimization algorithm: Harmony Search. In *Simulation, Vol. 76(2)*, pages 60–68, 2001.

- [46] Z. W. Geem, J. H. Kim, and G. V. Loganathan. Harmony Search optimization: application to pipe network design. In *Int. J. Model. Simulation*, Vol. 22(2), pages 125–133, 2002.
- [47] Z. W. Geem, K. S. Lee, and Y. Park. Application of Harmony Search to vehicle routing. In *American Journal of Applied Sciences*, Vol. 2(12), pages 1552–1557, 2005.
- [48] A. Genin and L. Karp. Effects of flow on competitive superiority in scleractinian corals. *Limnology and Oceanography*, vol. 39(4), pages 913–924, 1994.
- [49] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, Vol. 13, pages 533–549, 1986.
- [50] F. Glover and M. Laguna. *Tabu search*. Kluwer Academic Publishers, Boston, 1997.
- [51] D. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.
- [52] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, 1991.
- [53] J. A. Franco Gómez. *Un algoritmo basado en la optimización por enjambre de partículas para el problema de asignación 3-D*. PhD thesis, Instituto Tecnológico de La Paz, La Paz, Baja California Sur, México, Septiembre 2011.
- [54] M. Hata. *Empirical formula for propagation loss in land mobile radio services*. IEEE Transactions on Vehicular Technology. Vol. VT-29 (3), pages 317–325, 1980.
- [55] J. H. Holland. *Adaptation in natural and artificial systems*. Press, Ann Harbor, Michigan, 1975.
- [56] X. Hu, R. C. Eberhart, and Y. Shi. Recent advances in particle swarm. In *Proceedings of IEEE Congress on Evolutionary Computation, Portland, Oregon, EEUU*, pages 90–97, 2004.
- [57] International Commission on Non-Ionizing Radiation Protection (ICNIRP). Guidelines for limiting exposure to time-varying electric, magnetic and electromagnetic fields (up to 300 GHz). *ICNIRP*, 2009.
- [58] T. L. James, E. C. Brown, K. B. Keeling. A hybrid grouping genetic algorithm for the cell formation problem. *Computers & Operations Research* Vol. 34(7), pages 2059–2079, 2007.
- [59] Y. X. Jin, H. Z. Cheng, J. Y. Yan, and L. Zhang. New discrete method for Particle Swarm Optimization and its application in transmission network expansion planning. *Electric Power Systems Research*, Vol. 77, num 3-4, pages 227–233, Marzo 2007.
- [60] A. Jüttner, A. Orbán, Z. Fiala. Two new algorithms for umts access network topology design. *Eur. J. Oper. Res.* Vol. 164, pages 456–474, 2005.
- [61] J. Kennedy. The particle swarm: social adaptation of knowledge. *IEEE International Conference on Evolutionary Computation*, pages 303–308, Abril 1997.
- [62] J. Kennedy. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the 1999 Conference on Evolutionary Computation*, pages 1931–1938, 1999.



- [63] J. Kennedy and R. Eberhart. Particle Swarm Optimization. *IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, vol. 4*, pages 1942–1948, 1995.
- [64] J. Kennedy and R. Mendes. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Transactions on Systems Man and Cybernetics, Vol. 36 (C)*, pages 515–519, Julio 2006.
- [65] J. Kennedy and R. Mendes. Population structure and particle swarm performance. *2002 World Congress on Computational Intelligence, Honolulu, HI, EEUU, Vol. 2*, pages 1671–1676, Mayo 2002.
- [66] D. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science. Vol. 220*, pages 671–680, 1983.
- [67] B. Krishnamachari, S. Wicker. Base station location optimization in cellular wireless networks using heuristic search algorithms. *Studies in Fuzziness and Soft Computing, Soft Computing in Communications, Springer, Vol. 136(2)*, pages 201–219, 2004.
- [68] K. S. Lee and Z. W. Geem. A new meta-heuristic algorithm for continuous engineering optimization: Harmony Search theory and practice. In *Computer methods in applied mechanics and engineering, Vol. 194*, pages 3902–3933, 2005.
- [69] C. J. Liao, C. T. Tseng, and P. Luarn. A discrete version of Particle Swarm Optimization flowshop scheduling problems. *Computers and Operations Research, Vol. 34(10)*, pages 3099–3111, Octubre 2007.
- [70] D. Lirman. Fragmentation in the branching coral *Acropora Palmata* (Lamarck): growth, survivorship and reproduction of colonies and fragments. *Journal of Experimental Marine Biology and Ecology, Vol. 251(1)*, pages 41–57, 2000.
- [71] M. Mahdavi, M. Fesanghary, and E. Damangir. An improved Harmony Search algorithm for solving optimizing problems. In *Applied Mathematics and Computation, Vol. 188*, pages 1567–1579, 2007.
- [72] G. Mateus, R. V. Franqueira. Model and heuristic for a generalized access network design problem. *Telecommun. Syst. Vol. 15*, pages 257–271, 2000.
- [73] D. Mora Melia. *Diseño de redes de distribución de agua mediante algoritmos evolutivos. Análisis de eficiencia*. PhD thesis, Departamento de Ingeniería Hidráulica y Medio Ambiente, Universidad Politécnica de Valencia, Julio 2012.
- [74] M. Mitchell. *An introduction to genetic algorithms*. MIT Press, 1998.
- [75] J. Moláček, M. Denny, and J. W. M. Bush. The fine art of surfacing: its efficacy in broadcast spawning. *Journal of Theoretical Biology, Vol. 294*, pages 40–47, February 2012.
- [76] M. G. H. Omran and M. Mahdavi. Global-best Harmony Search. In *Applied Mathematics and Computation, Vol. 198(2)*, pages 643–656, 2008.
- [77] Organización Mundial de la Salud. Electromagnetic fields and public health: mobile phones. Fact sheet number 193, Junio, 2011. Enlace consultado en abril 2013.

- [78] I.H. Osman and G. Laporte. Metaheuristics: a bibliography. In *Annals of Operations Research* 63, pages 513–623, 1996.
- [79] Q. K. Pan, M. F. Tasgentiren, and Y. C. Liang. A discrete Particle Swarm Optimization algorithm for the no-wait flowshop scheduling problem. *Computer Operations Research*, Vol. 35(9), pages 2807–2839, Septiembre 2008.
- [80] W. Pang, K. P. Wang, C. G. Zhou, and L. J. Dong. Fuzzy discrete Particle Swarm Optimization for solving traveling salesman problem. In *Proceedings of the 4th International Conference on Computation and Information Technology (CIT)*, pages 796–800, 2004.
- [81] W. Pang, K. P. Wang, C. G. Zhou, L. J. Dong, M. Liu, H. Y. Zhang, and J. Y. Wang. Modified Particle Swarm Optimization based on space and transformation for solving traveling salesman problem. In *Proceedings fo the 3rd International Conference on Machine Learning and Cybernetics*, pages 2342–2348, 2004.
- [82] A. M. Pérez Bellido. *Mejora de algoritmos evolutivos en problemas de búsqueda de árboles óptimos: nuevos operadores sobre la codificación Dandelion*. PhD thesis, Escuela Politécnica Superior, Departamento de Teoría de la Señal y Comunicaciones. Universidad de Alcalá, Octubre 2010.
- [83] Y. Pomerleau, S. Chamberland, G. Pesant. A constraint programming approach for the design problem of cellular wireless networks. In *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering, Canada*, pages 881–884, 2003.
- [84] A. Quintero, S. Pierre. Assigning cells to switches in cellular mobile networks: a comparative study. *Computer Communications* Vol. 26, pages 950–960, 2003.
- [85] M.H. Repacholi. Environmental Health Criteria 137: electromagnetic fields (300 Hz to 300 GHz). *World Health Organisation (WHO/UNEP/IRPA)*, Geneva, 1993.
- [86] M. Riis. Deployment of mobile switching centers in a telecommunications network: a stochastic programming approach. *Telecommunication Systems* Vol. 26(1), pages 93–109, 2004.
- [87] E. Rosenberg. Dual ascent for uncapacitated telecommunications network design with access, backbone, and switch costs. *Telecommun. Syst* Vol. 16(3), pages 423–435, 2001.
- [88] S. Salcedo-Sanz, J. del Ser, S. Gil-López, I. Landa-Torres, and A. Portilla-Figueras. The Coral Reefs Optimization algorithm: an efficient meta-heuristic for solving hard optimization problems. *15th Applied Stochastic Models and Data Analysis International Conference, Mataró, España, Junio*, 2013.
- [89] S. Salcedo-Sanz, A. Pastor-Sánchez, D. Gallo-Marazuela, and A. Portilla-Figueras. A novel Coral Reefs Optimization algorithm for multi-objective problems. *The 14th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'2013), Heifei, China, Octubre*, 2013.
- [90] S. Salcedo-Sanz, X. Yao. A hybrid hopfield network-genetic algorithm for the terminal assignment problem. *IEEE Trans. Syst. Man Cybern. B* Vol. 34(6), pages 2343–2353, 2004.

- [91] A. Salman, I. Ahmad, and S. Al Madani. Particle Swarm Optimization for task assignment problem. *Microprocessors Microsystems*, Vol. 26(8), pages 363–371, 2002.
- [92] D. Y. Sha and C. Hsu. A hybrid Particle Swarm Optimization for job shop scheduling problem. *Computer and Industrial Engineering*, Vol. 51, pages 791–808, Diciembre 2006.
- [93] B. Shen, M. Yao, and W. Yi. Heuristic information based improved fuzzy discrete PSO method for solving TSP. In *Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, pages 859–863, 2006.
- [94] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, and Q. X. Wang. Particle Swarm Optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters*, Vol. 103(5), pages 169–176, Agosto 2007.
- [95] Y. Shi and R. C. Eberhart. A modified particle swarm optimizer. In *Proceedings of the 1998 IEEE Congress on Evolutionary Computation, Anchorage, AK*, 1998.
- [96] Y. Shi and R. C. Eberhart. Parameter selection in Particle Swarm Optimization. *Evolutionary Programming VIZ, New York: Springer Verlag*, pages 591–600, 1998.
- [97] Y. Shi and R. C. Eberhart. Empirical study of Particle Swarm Optimization. In *IEEE Congress on Evolutionary Computation*, pages 1945–1950, 1999.
- [98] S. Soni, S. Narasimhan, L. LeBlanc. Telecommunication access network design with reliability constraints. *IEEE Trans. Reliability* Vol. 53(4), pages 532–541, 2004.
- [99] J. L. Sumich. An introduction to the biology of marine life. pages 255–269, 1996.
- [100] Y. C. Tay, J. R. Guest, L. M. Chou, and P. A. Todd. Vertical distribution and settlement competencies in broadcast spawning coral larvae: implications for dispersal models. *Journal of Experimental Marine Biology and Ecology*, Vol. 409, num. 1-2, pages 324–330, 2011.
- [101] C. T. Tseng and C. J. Liao. A discrete Particle Swarm Optimization for lot-streaming flowshop scheduling problem. *European Journal of Operational Research*, Vol. 191(2), pages 360–373, Diciembre 2008.
- [102] J. C. Villela Tinoco. *Mecanismo de selección y control de una hiperheurística basada en evolución diferencial para optimización en espacios restringidos*. PhD thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, México, Octubre 2010.
- [103] K. P. Wang, L. Huang, C. G. Zhou, and W. Pang. Particle Swarm Optimization for traveling salesman problem. In *Proceedings of the 2nd International Conference on Machine Learning and Cybernetics*, pages 1583–1585, 2003.
- [104] Y. Wang, Y. Wanga, X. Y. Fenga, Y. X. Huanga, D. B. Pub, W. G. Zhoua, Y. C. Lianga, and C. G. Zhoua. A novel quantum swarm evolutionary algorithm and its applications. *Neurocomputing*, Vol. 70, num. 4-6, pages 633–640, Enero 2007.
- [105] A. Wetzel. *Evaluation of the effectiveness of genetic algorithms in combinational optimization*. unpublished, University of Pittsburgh, Pittsburgh, 1983.

- [106] D. Whitley. The GENITOR algorithm and selection pressure: why rank-based allocation of reproductive trials is best. In *Proceedings of the Third International Conference on Genetic Algorithms. San Mateo, California*, pages 116–121. Morgan Kaufmann, July 1989.
- [107] H. Yamashiro and M. Nishihira. Experimental study of growth and asexual reproduction in *Diaseris distorta* (Michelin, 1843), a free-living fungiid coral. *Journal of Experimental Marine Biology and Ecology*, Vol. 225(2), pages 253–267, 1998.
- [108] X. S. Yang. Harmony Search as metaheuristic algorithm. In *Music-Inspired Harmony Search Algorithm: Theory and Applications, Studies in Computational Intelligence. Springer, Vol. 191*, pages 1–14, 2009.
- [109] X. Yao, F. Wang, K. Padmanabham, S. Salcedo-Sanz. Hybrid evolutionary approaches to terminal assignment in communication networks. *IW. Hart, N. Krasnogor, J.E. Smith (Eds.), Recent Advances in Memetic Algorithms and Related Search Technologies, Springer-Verlag*, pages 129–159, 2004.
- [110] L. A. Zadeh. Soft Computing and Fuzzy Logic. *IEEE Software*, Vol. 11(6), pages 48–56, 1994.